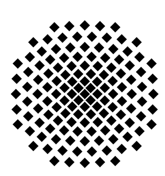
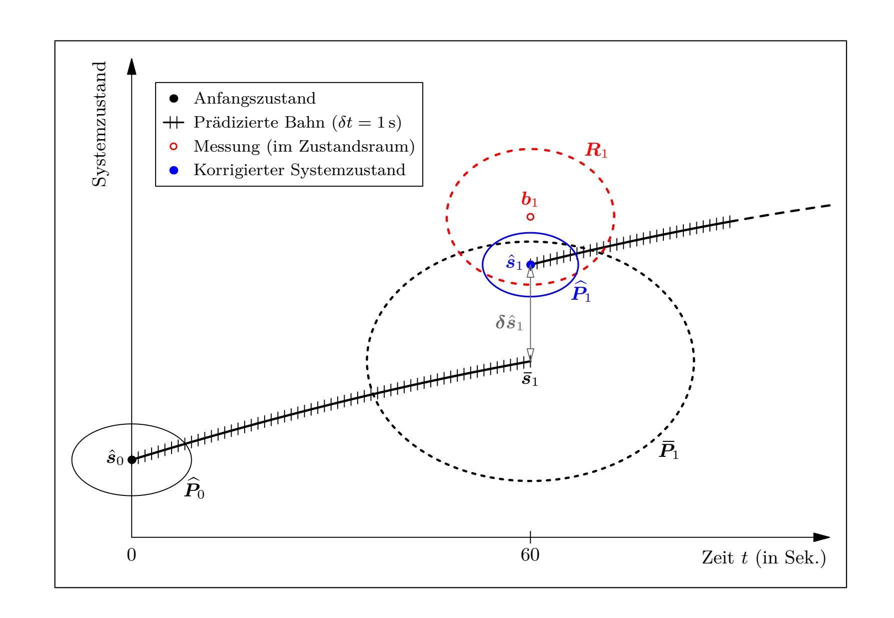


Universität Stuttgart

Geodätisches Institut



Orbitverdichtung mittels Kalman-Filterung am Beispiel der Satellitenmission GRACE



Diplomarbeit im Studiengang

Geodäsie und Geoinformatik

an der Universität Stuttgart

Benedikt Leinss

Stuttgart, Oktober 2011

Betreuer: Dr. Matthias Weigelt
Universität Stuttgart

Prof. Dr. sc. techn. Wolfgang Keller
Universität Stuttgart

Erklärung der Urheberschaft

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig, d.h. ohne Mitwirkung Dritter und ohne Verwendung anderer als der darin genannten Hilfsmittel erstellt habe.

Gedankengut, welches nicht von mir selbst stammt, ist stets durch die entsprechende Quellen- bzw. Literaturangabe gekennzeichnet.

Diese Arbeit wurde vormals in identischer oder ähnlicher Form weder veröffentlicht noch bei einem anderen Prüfungsamt eingereicht.

Stuttgart, den 4. Oktober 2011

Benedikt Leinss

Zusammenfassung

Das übergeordnete Thema der vorliegenden Arbeit ist die Verdichtung von Satellitenorbits. Grundsätzlich besteht hierbei die Aufgabe darin, für einen gegebenen Orbit eine korrespondierende Satellitenbahn mit einer höheren zeitlichen Auflösung zu bestimmen.

Eine solche Orbitverdichtung ist etwa bei einigen Ansätzen zur Schwerefeldbestimmung aus Level-1B-Daten der Satellitenmission GRACE nötig, weil dabei die GRACE-Navigationslösungen mit den K-Band-Beobachtungen zu kombinieren sind und sich diese Datensätze gerade hinsichtlich ihrer zeitlichen Abtastung unterscheiden.

Angesichts dieser Situation wird hier eine Orbitverdichtung mittels der erweiterten Kalman-Filterung (EKF) konzipiert und in einer MATLAB*-Toolbox umgesetzt. Neben einer Funktion, mit der sich prinzipiell weitgehend beliebige Orbits verdichten lassen, wird außerdem ein Filter entworfen, das einzig für GRACE ausgelegt ist und es ermöglicht, auch die hochgenauen K-Band-Relativgeschwindigkeiten in die Verdichtung einzubeziehen.

Um die Systemzustände und die zugehörigen Kovarianzen zu prädizieren, werden unterschiedliche Orbitintegrationen realisiert. Die meisten davon basieren auf der numerischen Integration von gewöhnlichen DGL-Systemen erster Ordnung in kartesischen, quasi-inertialen Koordinaten. Der hier komplexeste Integrator trägt den Störbeschleunigungen aufgrund der Anisotropie des Erdgravitationsfelds Rechnung, indem geeignete Gravitationsfeldmodelle genutzt werden.

Sämtliche Orbitintegratoren und die Routinen zur EKF-Orbitverdichtung werden mithilfe simulierter Daten durch Tests verifiziert. Außerdem vergleicht man die verschiedenen Prädiktionsmethoden miteinander im Hinblick auf die erzielten Genauigkeiten und deren Berechnungseffizienz.

Bei der gegen Ende durchgeführten Prozessierung realer GRACE-Daten zeigt sich, wie auch bei den vorherigen Tests, dass es dank der Kalman-Filterung gelungen ist, verdichtete GRACE-Orbits zu berechnen, welche erheblich besser zu den gemessenen K-Band-Relativgeschwindigkeiten passen als die gegebenen Navigationslösungen.

Abstract

The major objective of the following study is the densification of satellite orbits. In general, the task is here to determine a corresponding trajectory with a higher time resolution for a given satellite orbit.

An orbit densification is needed for example in some approaches dealing with gravity field recovery from level-1B-data of the satellite mission GRACE. These approaches involve combining the GRACE navigation solution with the K-band measurements. However, these data records have different sampling rates.

In view of this situation, an orbit densification by means of extended Kalman filtering (EKF) is conceived here and implemented in a MATLAB† toolbox. Adjacent to a function basically allowing to densify almost any kind of satellite orbit, a filter is designed exclusively for GRACE, which makes it possible to include the highly accurate K-band range rates in the densification.

* MATLAB® ist ein Software-Produkt des Unternehmens „The MathWorks, Inc.“. Es handelt sich um „eine hochentwickelte Sprache für technische Berechnungen und eine interaktive Umgebung für die Algorithmenentwicklung, die Visualisierung und Analyse von Daten sowie für numerische Berechnungen“ (vgl. www.mathworks.de/products/matlab/description1.html).

† MATLAB® is a software product of the company „The MathWorks, Inc.“. It's a „high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation“ (cf. www.mathworks.com/products/matlab/description1.html).

In order to predict future orbit states and the associated error covariances, different methods of orbit integrations are realized. Most of them are based on a numerical integration of a system of first order ordinary differential equations in cartesian Earth-centered inertial coordinates. The most complex integrator in place accounts for the disturbing accelerations caused by the anisotropic part of the gravitational field of the Earth by using appropriate gravity field models.

All orbit integrators and routines for the EKF orbit densification are verified by means of simulated data. Moreover, the different prediction methods are compared with each other regarding the achieved precisions and their computational efficiency.

In the penultimate chapter the processing of real GRACE data reveals, that owing to the Kalman filtering densified GRACE orbits could be calculated which fit the measured K-band range rates considerably better than the given navigation solutions.

Danksagung

Die Erstellung des vorliegenden Werks wäre mir in dieser Form nicht möglich gewesen ohne die Unterstützung durch folgende Personen, bei denen ich mich deshalb herzlich bedanken möchte:

- Herrn Dr. Matthias Weigelt für die hilfsbereite, stets interessierte und motivierende Betreuung, die Bereitstellung der realen GRACE-Datensätze und einiger Matlab-Funktionen.
- Herrn Prof. Dr. sc. techn. Wolfgang Keller für die hilfreichen Anregungen und die Prüfung dieser Arbeit.
- Herrn Dr. William I. Bertiger, der mir freundlicherweise einige Auskünfe über die am JPL produzierten GNV1B-Daten gab und mir außerdem einen Vorabdruck des Artikels (Bertiger et al., 2010) mit wissenswerten Hintergrundinformationen zukommen ließ.

Schließlich möchte ich mich besonders herzlich bedanken bei meinem Neffen für die kreative Gestaltung der Umschlagrückseite, sowie bei meinen Eltern und meiner Schwester, die mich im Studium auf ihre persönliche Weise unterstützen.

Vermerk über die zu dieser Arbeit gehörenden digitalen Daten

Der gedruckten Ausgabe dieses Dokuments liegt eine CD-ROM bei. Darauf sind die erstellte Toolbox sowie weitere Dateien gespeichert, auf die in den folgenden Kapiteln Bezug genommen wird.

Um die Toolbox in Matlab zu nutzen, sollte das Verzeichnis `\ekf_orbit_toolbox` mit dem gesamten Inhalt in ein lokales Festplattenverzeichnis kopiert werden, wobei darauf zu achten ist, dass die relative Verzeichnisstruktur erhalten bleibt (siehe auch `README.txt` auf der CD). In der Datei `contents.txt`, die man in diesem Verzeichnis findet, sind weitere Hinweise zur Verwendung der Toolbox gegeben und alle in der Toolbox enthaltenen Programme und Funktionen samt deren jeweiliger Aufgabe aufgelistet.

Der elektronischen Ausgabe dieses Dokuments ist die besagte Textdatei `contents.txt` als Dateianlage beigefügt. Sie lässt sich aus der PDF-Datei etwa mit dem Adobe® Acrobat® Reader™ ab Version 6 extrahieren oder mit der frei verfügbaren Software `Pdftk`[1].

1 The PDF Toolkit (GNU GPL), http://www.pdflabs.com/docs/install-pdftk/

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

AWP	Anfangswertproblem [S. 22, 23, 35–38, 40, 70]
CHAMP	Challenging Minisatellite Payload [S. 5–8]
DCM	Direction cosine matrix [S. 143, 145–147]
DGL	Differentialgleichung [S. V, 18, 21, 22, 25, 31, 33, 36–41, 43, 51, 53, 54, 70, 72, 73, 75–78, 87, 90, 133, 145]
DLR	Deutsches Zentrum für Luft- und Raumfahrt [S. 5]
EADS	European Aeronautic Defence and Space Company [S. 5]
EGM	Earth gravity model [S. 47, 52, 101]
EIGEN	European Improved Gravity Model of the Earth by new Techniques [S. 6, 7]
EKF	Erweitertes Kalman-Filter (engl.: „Extended Kalman Filter“) [S. V, VI, 26, 28, 33–35, 37, 38, 61, 96, 133–136, 138]
GFZ	Geoforschungszentrum Potsdam [S. 7]
GGM	GRACE Gravity Model [S. 7, 15, 47, 91, 92, 99]
GIS	Geodätisches Institut der Universität Stuttgart [S. 8]
GOCE	Gravity Field and Steady-State Ocean Circulation Explorer [S. 10–12]
GPS	Global Positioning System [S. 1, 5–8, 13, 14, 65, 99]
GPU	Graphics Processing Unit (Grafikprozessor(einheit) [eines Computers]) [S. 134]
GRACE	Gravity Recovery And Climate Experiment [S. V, VI, 1–3, 5–15, 21, 57–59, 62–64, 66, 72, 73, 80, 91, 92, 94, 96, 99, 100, 103, 113, 119, 122, 131, 132, 135–138, 151]
GSOC	German Space Operations Center [S. 7]
GUI	Graphical user interface (grafische Benutzerschnittstelle) [S. 64, 103, 108, 122, 125, 135]
hl-SST	High-low satellite-to-satellite tracking [S. 6, 8–10]
HSTX	Hughes STX Corporation (Lanham, Maryland) [S. 47]
IERS	International Earth Rotation and Reference Systems Service [S. 32, 47, 50, 86, 99]
ISDC	Information System and Data Center for geoscientific data [S. 7]
ITG	Institut für theoretische Geodäsie (Universität Bonn) [S. 8]
JGM	Joint gravity model [S. 47, 89]
JPL	Jet Propulsion Laboratory (Pasadena, Kalifornien) [S. VI, 7, 99, 132]
LEO	Low earth orbit [S. 5, 6, 21, 35, 132]
LKF	Linearisiertes Kalman-Filter (engl.: „Linearized Kalman Filter“) [S. 35, 43]
ll-SST	Low-low satellite-to-satellite tracking [S. 7–9]
LOS	Line of sight [S. 8, 9, 11, 12, 101, 132, 138]
MJD	Modifiziertes julianisches Datum [S. 63, 64, 66, 139]
NASA	National Aeronautics and Space Administration [S. 5, 47]
NGA	National Geospatial-Intelligence Agency [S. 47]
NIMA	National Imagery and Mapping Agency [S. 47]
OSU	Ohio State University [S. 47]
PDF	Portable Document Format (Adobe Systems Incorporated, www.adobe.com/de) [S. VI]
PO.DAAC	Physical Oceanography Distributed Active Data Center [S. 7, 99]
POD	Precise orbit determination [S. 6]
RDC	Raw Data Center [S. 7]
RWP	Randwertproblem [S. 21, 125, 128]
SDS	Science Data System [S. 7]
SGG	Satellitengravitationsgradiometrie (engl.: „satellite gravity gradiometry“) [S. 10, 96, 132]
SLR	Satellite laser ranging [S. 6]
SRF	Science reference frame [S. 132]
SST	Satellite-to-satellite tracking [S. 8]
TEC	Total electron content [S. 5]
UKF	„Unscented Kalman Filter“ [S. 133, 134]
UTCSR	University of Texas (at Austin), Center for Space Research [S. 7, 47]
WGS 84	World Geodetic System 1984 [S. 139, 143, 144]

Wichtige Bezeichnungen und Symbole

$a, b, \ldots$	Skalare
$\alpha, \beta, \ldots$	Winkelgrößen oder andere Skalare
$\boldsymbol{e}, \ldots;\ \boldsymbol{A}, \ldots$	Vektoren bzw. Matrizen
$\mathbf{0}$	Nullmatrix
$\mathbf{I}$	Einheitsmatrix
a	Große Halbachse der Kepler-Ellipse
a_{E}	Große Halbachse des Erdellipsoidmodells
$\boldsymbol{b}$	Mess-/Beobachtungsvektor
E	Exzentrische Anomalie
e	Exzentrizität
$\boldsymbol{e}_{12}$	Einheitsvektor in Richtung der Verbindungslinie zwischen den GRACE-Satelliten
$\boldsymbol{F}$	Dynamik-/Systemmatrix
$\boldsymbol{f}_{\mathrm{K}}$	nicht-lineare Kräftefunktion (deterministisch modellierte Beschleunigungen)
$\boldsymbol{G}$	Einflussmatrix des Systemrauschens (Störgrößenmatrix)
GM	Geozentrische Gravitationskonstante
GMST	Greenwich Mean Sidereal Time (mittlere Sternzeit von Greenwich)
$\boldsymbol{H}$	Messmatrix
i	Inklination
$\boldsymbol{K}$	Kalman-Verstärkungs-/Gewichtsmatrix
M	Mittlere Anomalie
n	Mittlere Bewegung
ω	Argument des Perigäums
ω_{E}	mittlere Winkelgeschwindigkeit der Erde
$\boldsymbol{\omega}$	Vektor der Winkelgeschwindigkeiten/Drehraten
Ω	Rektaszension des aufsteigenden Knotens
$\boldsymbol{\Omega}$	Cartan-Matrix
$\mathcal{O}$	Landau-Symbol für Ordnung bzw. obere Schranke einer Funktion
$\boldsymbol{P}$	Kovarianzmatrix des Zustandsvektors
$\boldsymbol{\Phi}$	Transitions-/System-/Zustandsübergangsmatrix
$\boldsymbol{Q}(t)$	Spektraldichtematrix des Systemrauschens
$\boldsymbol{Q}_k$	Kovarianzmatrix des Systemrauschens
$\boldsymbol{R}$	Kovarianzmatrix des Messrauschens od. Rotationsmatrix
R_{E}	Äquatorradius der Erde
$\rho,\ \dot{\rho},\ \ddot{\rho}$	Relativentfernung, -geschwindigkeit, -beschleunigung bezüglich $\boldsymbol{e}_{12}$
$r,\ \varphi,\ \lambda$	sphärische, erdfeste Koordinaten (Radius, Breite, Länge)
$\boldsymbol{s}$	Zustandsvektor
σ_a	Standardabweichung der Größe a
σ_{ab}	Kreuzkorrelation zwischen der Größe a und b
t	Zeit
$\boldsymbol{u}$	Vektor deterministischer Eingangsgrößen (Stellgrößen)
$\boldsymbol{v}$	Messrauschen
$\boldsymbol{w}$	Vektor stochastischer Eingangsgrößen (Störgrößen, weißes Systemrauschen)
$\boldsymbol{x},\ \dot{\boldsymbol{x}},\ \ddot{\boldsymbol{x}}$	Positions-, Geschwindigkeits- und Beschleunigungsvektor im geozentrisch gelagerten, erdfesten Bezugssystem
$\boldsymbol{X},\ \dot{\boldsymbol{X}},\ \ddot{\boldsymbol{X}}$	Positions-, Geschwindigkeits- und Beschleunigungsvektor im geozentrisch gelagerten, raumfesten Bezugssystem

1

Kapitel 1

Prolog

Entsprechend der klassischen Definition der Geodäsie als „*Wissenschaft von der Ausmessung und Abbildung der Erdoberfläche*“ (Friedrich Robert Helmert, 1880) gehört die Bestimmung der geometrischen Figur und des Schwerefelds der Erde zu den grundlegenden Aufgaben der Geodäsie. Die genaue Kenntnis des Erdschwerefelds ist in vielen wissenschaftlichen Disziplinen, wie etwa der Ozeanographie und der Geophysik, aber auch in der Geodäsie selbst von enormer Bedeutung. So sind die meisten geodätischen Beobachtungen an der Erdoberfläche (wie z.B. geometrische Nivellements) stets an der jeweiligen Lotrichtung orientiert, d.h. sie sind vom Erdschwerefeld abhängig und müssen auf geeignete physikalische Bezugsflächen, wie etwa das Geoid als Niveaufläche[1], reduziert werden, um physikalisch korrekte, einheitliche und vergleichbare Messungen (Höhendifferenzen) zu erhalten.

Vor der zweiten Hälfte des 20. Jahrhunderts konnte das Erdschwerefeld nie vollständig, sondern nur lokal auf den Kontinenten in ausreichender Genauigkeit mithilfe der (bodengebundenen) Gravimetrie sowie durch Messung von Lotabweichungen erfasst werden. Erst danach, im Zeitalter der künstlichen Erdsatelliten, konnten einige Satellitenmissionen verwirklicht werden, die eine globale Schwerefeldbestimmung mit wirtschaftlich vertretbarem Aufwand ermöglicht haben. Mit den dazu erforderlichen Beobachtungs- und Auswerteverfahren befasst sich die Satellitengeodäsie.

Aufgrund günstiger Beobachtungskonstellationen, höherer Datendichten und Messgenauigkeiten sind heute sogar die zeitlichen Veränderungen des Schwerefelds detektierbar, so auch bei der Satellitenmission GRACE.

Satelliten können als Prüfkörper aufgefasst werden, die sich im ‚freien Fall‘ um die Erde bewegen und dabei unter anderem dem Einfluss des Erdschwerefelds ausgesetzt sind. Man beobachtet in der Regel die Bahn des Satelliten und gegebenenfalls die Relativbewegung zwischen/zu anderen Prüfmassen. Für das inverse Problem, wie aus der Beobachtung derartiger Bewegungen die das Schwerefeld repräsentierenden Parameter selbst abgeleitet werden können, haben bereits zahlreiche Forscher adäquate Lösungen erarbeitet[2].

1.1 Problemstellung

Bei den meisten Techniken zur Schwerefeldbestimmung, die bislang für die GRACE-Mission entwickelt wurden, sind unterschiedliche Beobachtungsgrößen zu kombinieren. Dabei handelt es sich zum einen um die sogenannten GPS-Navigationslösungen der beiden GRACE-Satelliten (Bahnvektoren bestehend aus Positions- und Geschwindigkeitskomponenten) und zum anderen um die aus dem K-Band-Entfernungsmesssystem gewonnenen Beobachtungen (relative Distanz ρ der Satelliten sowie zeitliche Änderungen dieser Entfernung). Das eigentliche Problem besteht nun darin, dass sich die entsprechenden realen, frei erhältlichen Daten dieser Beobachtungsgrößen (sog. „Level-1B-Daten“) hinsichtlich ihrer Genauigkeit und zeitlichen Auflösung unterscheiden.

1 Zur ausführlichen Definition des Geoids siehe z.B. Grafarend (1994).
2 Z.B. O'Keefe (1957); Rummel et al. (1978); Ilk (1983); Colombo (1984); Jekeli et al. (1999); Austen und Reubelt (2000); Mayer-Gürr (2006)

Während die K-Band-Messungen ρ mit einer Genauigkeit von $< 10\,\mu m$ in Zeitintervallen von fünf Sekunden vorliegen, stehen die Navigationslösungen nur alle 60 Sekunden mit Genauigkeiten von durchschnittlich etwa 2 cm bei den Positions- und $0.02\,\mathrm{mm/s}$ bei den Geschwindigkeitskomponenten zur Verfügung.

In einer Fehlerabschätzung stellt sich heraus, dass die aus den Navigationslösungen beider Satelliten ermittelte Relativgeschwindigkeit $\|\dot{\boldsymbol{X}}_{12}\|$ die kritischste Größe bei der Schwerefeldlösung (nach dem Ansatz der Gravitationsgradiometrie) darstellt und auf wenigstens $34\,\mu\mathrm{m/s}$ genau bekannt sein muss, um eine Geoidhöhengenauigkeit von einem Milli-Eötvös zu gewährleisten. Diese Genauigkeitsanforderung erfüllen die Navigationslösungen jedoch keineswegs.

Für die Schwerefeldlösung bedeutet dies in der Konsequenz, dass deren Gesamtqualität aufgrund der ungenauen Navigationslösungen schlechter ist als sie eigentlich dank der hochgenauen K-Band-Beobachtungen sein könnte, d.h. das Genauigkeitspotential der GRACE-Schwerefeldlösung kann wegen der minder genauen Navigationslösungen nicht ausgeschöpft werden.

1.2 Zielsetzung und Methodik

Offensichtlich ist es notwenig, die Navigationslösungen auf eine zeitliche Auflösung von fünf Sekunden oder weniger zu verdichten, weil ansonsten viele K-Band-Beobachtungen bei einer Gravitationsfeldbestimmung unberücksichtigt blieben. Allerdings zeigt sich, dass einfache Interpolations- und Orbitintegrationsmethoden, welche als Eingangsgrößen einzig den zu verdichtenden Orbit nutzen, an den gestellten hohen Anforderungen scheitern, da sie nicht die gewünschte Genauigkeitssteigerung gegenüber der Navigationslösung liefern.

Anhand der Geometrie des GRACE-Systems lässt sich ein funktionaler Zusammenhang zwischen der kritischen Größe $\|\dot{\boldsymbol{X}}_{12}\|$ und der K-Band-Relativgeschwindigkeit $\dot{\rho}$ erkennen und ableiten. Die Idee bzw. das Ziel dieser Arbeit ist daher, eine Verdichtung mittels Orbitintegration umzusetzen, bei der diese Relation zu den hochgenauen K-Band-Messungen $\dot{\rho}$ gewissermaßen als Zwangsbedingung miteinfließt, um auf diese Weise möglichst eine gegenüber der Navigationslösung gesteigerte Genauigkeit der verdichteten Orbits in Bezug auf die gemessenen K-Band-Relativgeschwindigkeiten $\dot{\rho}$ und damit auch in Bezug auf die Größe $\|\dot{\boldsymbol{X}}_{12}\|$ zu erreichen.

Neben anderen, vermutlich ebenfalls zielführenden Methoden aus der Schätztheorie, wie z.B. der sequentiellen Ausgleichung, wurde für diese Arbeit die (erweiterte) Kalman-Filterung ausersehen. Es wurden zwei verschiedene Filtervarianten zur Orbitverdichtung konzipiert und in MATLAB umgesetzt. Das erste Filter verwendet nur die Bahnvektoren *eines* gegebenen, unverdichteten Orbits als Beobachtungen und taugt somit zur Verdichtung einzelner, beliebiger (im raumfesten geozentrischen System vorliegender) Satellitenorbits. Die zweite Filtervariante wurde speziell für den beschriebenen Sachverhalt bei GRACE entworfen. Diese verdichtet die beiden gegebenen Navigationslösungen quasi gleichzeitig und berücksichtigt zusätzlich zu den beiden Bahnvektoren der Navigationslösungen auch die (einzelnen) K-Band-Messungen $\dot{\rho}$, um die mittels Orbitintegration prädizierten Systemzustände zu korrigieren. Im Gegensatz zur sequentiellen Ausgleichung kann hier bei der Kalman-Filterung ein Modell für das Systemrauschen eingeführt werden, das den Unsicherheiten bei der Orbitintegration Rechnung trägt.

1.3 Agenda

Im zweiten Kapitel werden zunächst neben der Satellitenmission GRACE auch einige der existierenden Ansätze zur GRACE-Schwerefeldbestimmung vorgestellt. Anschließend wird darin die erwähnte Fehlerabschätzung und eine Voruntersuchung einfacher Techniken zur Orbitverdichtung

mithilfe simulierter Daten durchgeführt. Mit dieser Voruntersuchung wird nachgewiesen, dass die einfachen Methoden nicht die geforderten Genauigkeiten liefern.

Das dritte Kapitel beginnt mit den theoretischen Grundlagen der linearen Kalman-Filterung. Danach geht man über zur erweiterten Kalman-Filterung im Anwendungsfall der Orbitverdichtung. Dabei werden Aspekte des Systemrauschens und der Filterstabilität diskutiert und die unterschiedlichen Kräftemodelle zur Prädiktion der Systemzustände und zugehörigen Kovarianzmatrizen beschrieben. Am Ende des Kapitels werden die Unterschiede beim Korrekturschritt der zwei realisierten Filtervarianten anhand der formelmäßigen Beziehungen der erforderlichen Größen dargelegt.

Die Bestandteile und die Implementierung der in MATLAB erstellten Toolbox zur Orbitverdichtung werden im vierten Kapitel erläutert. Im Anschluss untersucht man mithilfe simulierter Daten in diversen Tests, ob die Orbitintegratoren korrekte und hinreichend genaue Ergebnisse liefern und ob die Prädiktion der Kovarianzmatrizen sowie die gesamte Orbitverdichtung zufriedenstellend funktioniert.

Das fünfte Kapitel beinhaltet Informationen zu den gegebenen realen GRACE-Daten und befasst sich nach einer Vorab-Analyse dieser Daten mit deren Prozessierung durch die erstellte Toolbox. Dabei werden die Ergebnisse der mit unterschiedlichen Parametern durchgeführten Verarbeitungen dargestellt, interpretiert und miteinander verglichen.

Im letzten Kapitel wird aufgrund der positiven Resultate der erfolgten Orbitverdichtungen ein Ausblick gegeben auf den logischen nächsten Schritt der Gravitationsfeldbestimmung. Ferner werden Verbesserungsmöglichkeiten im Hinblick auf das Kräftemodell und die umgesetzte Toolbox genannt und Empfehlungen für zukünftige Arbeiten ausgesprochen.

2 Kapitel 2 Umfassende Einführung und Voruntersuchungen

2.1 Satellitenmission GRACE

Die im Jahre 2002 gestartete Satellitenmission GRACE (Gravity Recovery And Climate Experiment) ist ein Gemeinschaftsprojekt der amerikanischen Weltraumorganisation NASA und des Deutschen Zentrums für Luft- und Raumfahrt (DLR). Das Hauptziel der Mission besteht darin, die mittleren und zeitvariablen Komponenten des Erdschwerefelds mit einer bis dato unerreichten Genauigkeit und zeitlich-räumlicher Auflösung über einen Missionszeitraum von 5 Jahren[1] zu erfassen und global zu modellieren (vgl. Parkinson et al., 2006, S. 158; Case et al., 2010, S. 7). Ein weiteres Missionsziel ist die regelmäßige Durchführung von sog. „limb-sounding“-Messungen zu GPS-Satelliten, die Rückschlüsse auf den Elektronengehalt (TEC) der Ionosphäre und auf die Temperaturverteilung sowie den Wasserdampfgehalt in der Atmosphäre ermöglichen. Derartige Informationen sind insbesondere für klimatologische Studien von Interesse.

2.1.1 Missionsparameter und Instrumentarium

Das Raumsegment der GRACE-Mission bilden zwei fast baugleiche Satelliten, die auf ungefähr gleichen, annähernd kreisförmigen Bahnen (Exzentrizität $e \approx 0.001$) hintereinander im Abstand von 220 km ± 50 km die Erde umrunden (Tandem- bzw. „follower-leader“-Konfiguration). Mit der sehr niedrigen Bahnhöhe von derzeit etwa 460 km kann man GRACE ebenso wie die Vorgängermission CHAMP (Challenging Minisatellite Payload) in die Kategorie der LEO-Satelliten (engl.: „low earth orbiting satellites“) einordnen, deren Bahnen im Gegensatz zu höheren Orbits besonders empfindlich gegenüber den Feinstrukturen des Erdschwerefelds sind. Durch die polnahe Bahn (Inklination $i \approx 89°$) und die kurze Umlaufdauer von etwa 94 Minuten (etwas über 15 Umläufe pro Tag) werden in kurzer Zeit weite Teile der Erdoberfläche erfasst.

Der Aufbau und das Instrumentarium der GRACE-Zwillingssatelliten ist abgesehen vom K-Band-Messsystem (siehe unten) weitgehend mit dem Vorläufer CHAMP identisch, der ebenfalls bei EADS Astrium in Friedrichshafen gebaut wurde. Die Nutzlast der GRACE-Kleinsatelliten umfasst u.a. jeweils:

- Zweifrequenz-*BlackJack*-GPS-Empfänger zur Positionsbestimmung und zum Abgleich interner Zeitgeber mit der GPS-Zeit
- hochpräzises Entfernungsmesssystem im Mikrowellen-K-/Ka-Band (*High Accuracy Inter-satellite Ranging System* HAIRS)
- Beschleunigungssensoren (*SuperStar Accelerometer* SSA) zur Erfassung von Oberflächenbeschleunigungen (nicht-gravitativen Störbeschleunigungen), die etwa durch Atmosphärenreibung, Sonnenstrahlungsdruck oder Erdalbedo zustande kommen

1 Erfreulicherweise wurde die fünfjährige Missionsdauer bereits überschritten und die Satelliten sind weiterhin in Betrieb

- Sternkameras (*Star Camera Assembly* SCA), die mithilfe von Sternkatalogen eine Überprüfung der Satellitenorientierung im Raum ermöglichen
- Laserreflektor-Einheit (*Laser Retro-Reflector* LRR), um eine von GPS unabhängige Verfolgung bzw. Positionsbestimmung des Satelliten von Bodenstationen aus mittels SLR-Messungen zu gewährleisten
- Taktfrequenzgeber (*Ultra-Stable Oscillator* USO), der eine stabile Grundfrequenz für HAIRS und SPU bereitstellt
- diverse Systeme zur Datenprozessierung (SPU/IPU: *Signal/Instrument Processing Unit*).

Eine detaillierte Beschreibung aller an Bord befindlichen Instrumente und Prozessierungseinheiten findet man etwa in Bertiger et al. (2003).

2.1.2 Satellite-to-Satellite-Tracking

Das bereits bei CHAMP realisierte Messkonzept des „Satellite-to-Satellite-Trackings“ in der „high-low“-Anordnung (hl-SST) findet auch bei GRACE Anwendung. Dabei werden permanent die Positionen und Geschwindigkeiten der LEO-Satelliten mittels GPS[1] aufgezeichnet und zusätzlich nicht-gravitative Störbeschleunigungen mithilfe der an Bord installierten Akzelerometer erfasst. In der hl-SST-Auswertung nutzt man die Akzelerometermessungen, um mit einem a priori Erdgravitationsfeld und Gezeitenmodellen eine Kräftefunktion zu bilden, die nach numerischer Integration einen Referenzorbit des Satelliten liefert (Verfahren der „Precise orbit determination“, POD). Die Abweichungen des Referenzorbits zu dem aus GPS-Messungen kinematisch bestimmten Orbit, d.h. die Bahnstörungen, sind Ausgangspunkt für die Schätzung der Kugelfunktionskoeffizienten des gesuchten Erdschwerefelds. Dieses Prinzip der hl-SST-Schwerefeldbestimmung ist beispielsweise in Reigber et al. (2003) skizziert. Der Nachteil hierbei ist, dass die Störungen der Satellitenbahn keinen direkten räumlichen Bezug zu den Schwerefeldanomalien haben. Die Bahnstörung zu einem bestimmten Zeitpunkt stellt vielmehr den Einfluss aller bereits zuvor auf den Satelliten eingewirkten Schwerefeldstörungen dar (Hess und Keller, 1999). Infolgedessen lassen sich die gesuchten Kugelfunktionskoeffizienten nur mithilfe fehlerbehafteter Hypothesen über deren Signalstärke eindeutig aus den Bahnstörungen auflösen und es können im Wesentlichen lediglich die langwelligen Strukturen des Gravitationsfelds signifikant bestimmt werden (beim Schwerefeldmodell EIGEN-CHAMP03S etwa bis zum sphärisch-harmonischen Grad $n = 60$, vgl. Reigber et al. (2005a)).

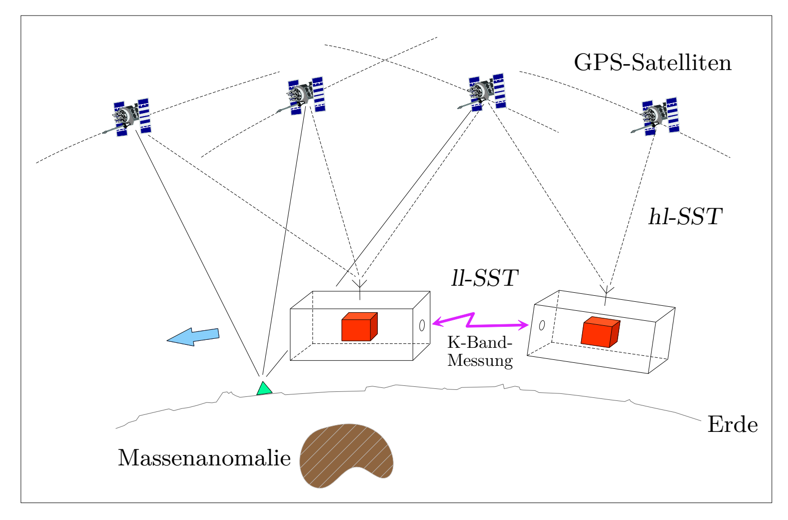


Abb. 2.1: Messkonzepte der GRACE-Satellitenmission (nach Balmino et al., 1999, S. 20)

1 GPS-Bahnhöhe ca. 20200 km

Um die erwähnten Nachteile aufzuwiegen und eine verbesserte Schwerefelddetektion zu erreichen, wurde bei GRACE erstmals das sogenannte „Low-low satellite-to-satellite tracking“ (ll-SST) verwirklicht. Das Grundprinzip dabei ist, dass Unregelmäßigkeiten im Erdschwerefeld Veränderungen des gegenseitigen Abstands der Satelliten verursachen. Da der führende GRACE-Satellit eine Massenanomalie – ein Gebiet höherer/geringerer Dichte (z.B. Gebirge/Tal) – früher als der nachfolgende Satellit überfliegt, erfährt der erste Satellit eine stärkere bzw. schwächere Beschleunigung als der nachfolgende, sodass folglich deren gegenseitige Distanz variiert. Aus diesem Grund wird mittels zweier Einweg-Mikrowellenmesssysteme permanent die Länge der Verbindungslinie zwischen den GRACE-Satelliten (Relativentfernung) und deren zeitliche Änderung (Relativgeschwindigkeit) nach dem Phasenvergleichsverfahren gemessen. Das System arbeitet dabei auf zwei Frequenzen im Mikrowellen K- und Ka-Band, um ionosphärische Einflüsse zu kompensieren. Bereits Wolff (1969) hat aufgezeigt, dass derartige Relativgeschwindigkeiten (im Gegensatz zu den Beobachtungen der hl-Konfiguration) direkt proportional zu einer Schwerefeldgröße sind, nämlich zu den Differenzen des Schwerepotentials zwischen den Satellitenpositionen. Prinzipiell sind die hochgenauen K-Band-Beobachtungen daher dazu geeignet, auch sehr kleine Variationen im Schwerefeld sichtbar zu machen.

2.1.3 Datenvorverarbeitung und -auswertung

Die Betreiber ordnen die GRACE-Daten je nach Prozessierungsstatus den Kategorien Level-0, Level-1A, Level-1B und Level-2 zu (Bettadpur, 2007). An zwei Standorten in Deutschland (GSOC in Weilheim, RDC in Neustrelitz) werden die GRACE-Rohdaten (Level-0) empfangen und archiviert. Die Vorverarbeitung und wissenschaftliche Auswertung der Daten erfolgt durch das *Science Data System* (SDS). Hierbei ist das *Jet Propulsion Laboratory* (JPL) in Pasadena, das *Center for Space Research* der Universität Texas (UTCSR) und das *Geoforschungszentrum Potsdam* (GFZ) beteiligt.

Zunächst werden die Level-0-Daten umformatiert, von System- in SI-Einheiten konvertiert, kalibriert und hinsichtlich ihrer Qualität gekennzeichnet. Die ursprünglichen Daten sind bis auf unlesbare bzw. fehlerhafte Datenpakete vollständig aus den so erhaltenen Level-1a-Daten wiederherstellbar. Danach folgt die nicht-umkehrbare Verarbeitung in den Level 1b, wobei das SDS u.a. die Messungen auf ein einheitliches Zeitmaß bezieht, mittels geeigneter Filter die Datendichte reduziert, Ausreißer eliminiert und Glättungen der Daten vornimmt. Außerdem werden weitere Datenprodukte berechnet, z.B. „präzise Ephemeriden der GRACE-Satellitenorbits“ (Bettadpur, 2007, S. 39 zum Produkt GNV1B). Die Level-2-Produkte beinhalten schließlich die aus den Level-1B- und weiteren Daten berechneten statischen und zeitvariablen Schwerefeldmodelle.

Nur die Level-1B- und Level-2-Daten sind öffentlich zugänglich und können einerseits nach einer Benutzerregistrierung über das Internetportal ISDC[1] des Geoforschungszentrums Potsdam (GFZ) und andererseits über das PO.DAAC[2] des JPL abgerufen werden. Die für diese Arbeit zur Verfügung gestellten realen GRACE-Daten beruhen auf Level-1B-Daten (K-Band-Produkt KBR1B, GPS-Navigationslösungen GNV1B), die aus dem Archiv des PO.DAAC bezogen wurden (nähere Informationen s. Abschn. 5.1).

Das ebenfalls am GFZ herausgegebene, vollständig aus GRACE-Beobachtungen abgeleitete Level-2-Produkt EIGEN-GRACE02S ist ein bis zum Grad 150 aufgelöstes Schwerefeldmodell. Laut Reigber et al. (2005b) weist es eine Geoidhöhengenauigkeit von 1 cm auf bei einer räumlichen Auflösung (halbe Wellenlänge) von 275 km (sphärisch-harmonischer Grad 75). Die GGM-Modelle des UTCSR erreichen ähnliche Auflösungen (s. Tapley et al., 2005). Gegenüber typischen CHAMP-Modellen hat sich damit die Auflösung um etwa eine Größenordnung verbessert.

1 http://isdc.gfz-potsdam.de
2 http://podaac.jpl.nasa.gov/gravity/grace

Die Methoden zur Bestimmung derartiger Gravitationsfeldmodelle in Form von Kugelfunktionskoeffizienten aus GRACE-Daten sind inzwischen zahlreich: neben der klassischen SST-Auswertung, die auf den Variationsgleichungen basiert (Colombo, 1984), nutzen andere Ansätze beispielsweise die Energiebilanzmethode (z.B. Jekeli, 1999) oder fassen kurze Bahnbögen als Randwertaufgabe in Form einer Integralgleichung auf. Der zuletzt genannte Integralgleichungsansatz sowie Erläuterungen zu verschiedenen Modellansätzen der Gravitationsfeldbestimmung aus SST-Beobachtungen sind in Mayer-Gürr (2006) aufgezeigt. Das nach diesem Ansatz am Institut für Geodäsie und Geoinformation der Universität Bonn abgeleitete Modell ITG-Grace03s bildet das statische Erdschwerefeld bis zum Grad 180 ab (Mayer-Gürr, 2007). Weiters ist der Beschleunigungsansatz zu nennen, der bereits erfolgreich am Geodätischen Institut der Universität Stuttgart (GIS) für die CHAMP-Mission erarbeitet und implementiert wurde (Austen und Reubelt, 2000; Reubelt, 2009). Im Fall der GRACE-Mission setzt man die zeitlichen Ableitungen der gemessenen Entfernungsänderungen (Relativbeschleunigungen) in Richtung der Verbindungslinie beider Satelliten (*line of sight*, LOS) in Bezug zur Differenz der vektoriellen Gravitationsbeschleunigungen (ebenfalls projiziert auf die LOS-Richtung). Letztere sind über die Newton-Eulersche-Bewegungsgleichung direkt mit Gradienten des Gravitationspotentials verknüpft, welche die Änderung der Gravitation in der LOS-Richtung beschreiben (z.B. Rummel et al., 1978; Austen und Grafarend, 2004).

Die meisten Techniken zur Bestimmung der Schwerefeldkoeffizienten aus GRACE-Daten – so auch der Beschleunigungsansatz – schöpfen das Genauigkeitspotential der K-Band-Messungen bislang nicht vollständig aus. Der Grund hierfür ist, dass die hochgenauen K-Band-Beobachtungen (ll-SST) mit den weniger genauen GPS-Beobachtungen (hl-SST) kombiniert werden müssen, da mit den K-Band-Messungen selbst eine absolute Positionierung im Raum nicht möglich ist, d.h. absolute Positionen bzw. Geschwindigkeiten der Satelliten sind jeweils aus den GPS-Messungen zu ermitteln, während die Schwerefeldinformation hauptsächlich den K-Band-Messungen entstammt. Diese Problematik ist gut anhand des funktionalen Zusammenhangs zwischen den Beobachtungen der hl- bzw. ll-SST-Konfiguration und den das Schwerefeld repräsentierenden Parametern ersichtlich, welcher daher im folgenden Abschnitt hergeleitet wird.

2.1.4 Geometrie des GRACE-Systems

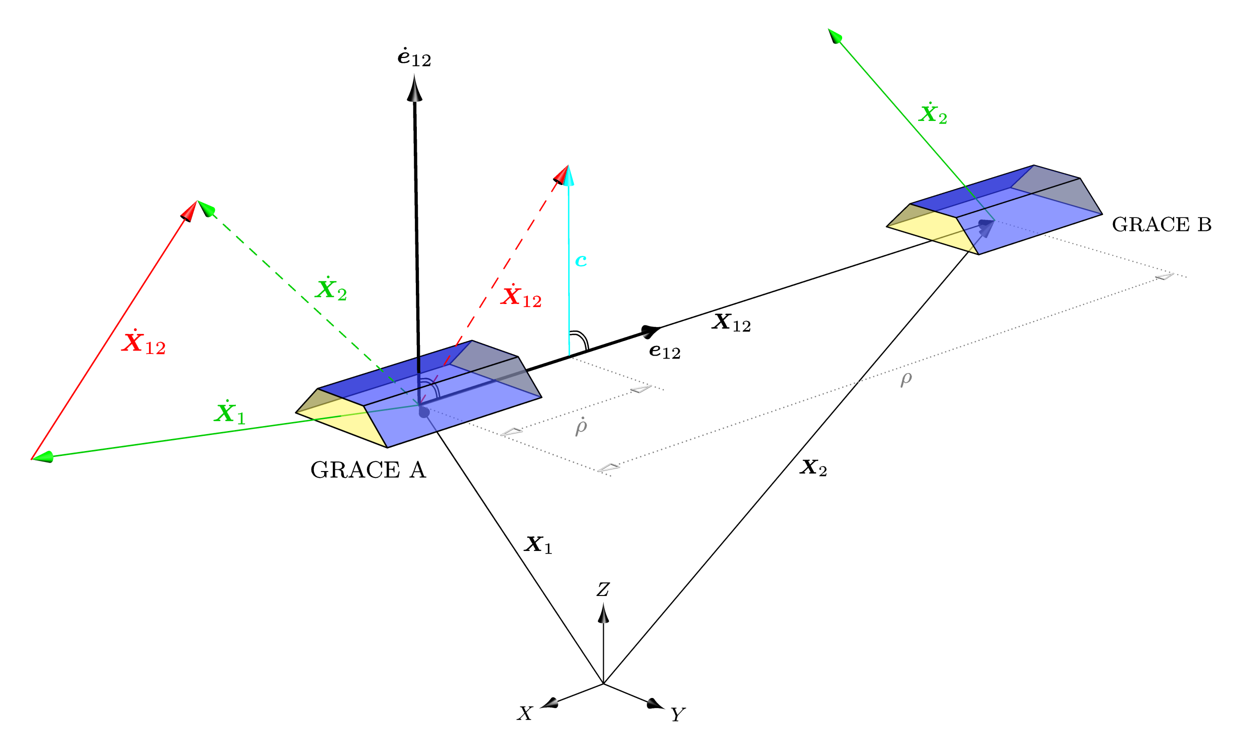


Abb. 2.2: Geometrische Interpretation wichtiger GRACE-Beobachtungen

Es seien $\boldsymbol{X}_1$ und $\boldsymbol{X}_2$ die kartesischen Ortsvektoren der beiden GRACE-Satelliten bezüglich des geozentrisch gelagerten, raumfesten Bezugssystems. Dann gilt für den gegenseitigen Abstand der Satelliten

$$\rho = \|\boldsymbol{X}_2 - \boldsymbol{X}_1\| = \|\boldsymbol{X}_{12}\| = \frac{\|\boldsymbol{X}_{12}\|^2}{\|\boldsymbol{X}_{12}\|} = \frac{\langle \boldsymbol{X}_{12}, \boldsymbol{X}_{12} \rangle}{\|\boldsymbol{X}_{12}\|} = \langle \boldsymbol{X}_{12}, \boldsymbol{e}_{12} \rangle, \tag{2.1}$$

wobei

$$\boldsymbol{e}_{12} = \frac{\boldsymbol{X}_{12}}{\|\boldsymbol{X}_{12}\|} = \frac{\boldsymbol{X}_{12}}{\rho} \tag{2.2}$$

den Einheitsvektor in Richtung der Verbindungslinie der Satelliten darstellt. Einmaliges Differenzieren nach der Zeit liefert die Relativgeschwindigkeit

$$\dot{\rho} = \langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle + \underbrace{\langle \boldsymbol{X}_{12}, \dot{\boldsymbol{e}}_{12} \rangle}_{=0, \text{ da } \boldsymbol{X}_{12} \perp \dot{\boldsymbol{e}}_{12}} = \langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle \tag{2.3}$$

und zweimaliges Ableiten führt zu dem Ausdruck

$$\ddot{\rho} = \langle \ddot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle + \langle \dot{\boldsymbol{X}}_{12}, \dot{\boldsymbol{e}}_{12} \rangle \tag{2.4}$$

für die Relativbeschleunigung. Für die Ableitung des Einheitsvektors gilt

$$\dot{\boldsymbol{e}}_{12} \overset{(2.2)}{=} \frac{\partial}{\partial t}\left(\frac{\boldsymbol{X}_{12}}{\rho}\right) = \frac{\dot{\boldsymbol{X}}_{12}}{\rho} - \frac{\dot{\rho}\boldsymbol{X}_{12}}{\rho^2} \overset{(2.2)}{=} \frac{1}{\rho}\underbrace{(\dot{\boldsymbol{X}}_{12} - \dot{\rho}\,\boldsymbol{e}_{12})}_{\boldsymbol{c}}. \tag{2.5}$$

Wie man Abbildung 2.2 entnehmen kann, ist offenbar die Relativgeschwindigkeit $\dot{\boldsymbol{X}}_{12}$ über den zur LOS-Richtung orthogonalen Vektor $\boldsymbol{c}$ mit der K-Band-Relativgeschwindigkeit $\dot{\rho}$ verknüpft (s. auch Rummel et al., 1978). Somit ist nun der bereits eingangs erwähnte Zusammenhang zwischen den beiden Größen $\dot{\boldsymbol{X}}_{12}$ und $\dot{\rho}$ einerseits geometrisch, und mit Gl. (2.3) andererseits auch mathematisch aufgezeigt.

Mit Gleichung (2.5) folgt für die Relativbeschleunigung

$$\ddot{\rho} = \langle \ddot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle + \frac{1}{\rho}\left(\|\dot{\boldsymbol{X}}_{12}\|^2 - \dot{\rho}^2\right). \tag{2.6}$$

Dies ist äquivalent zu der Beziehung (vgl. Keller und Sharifi, 2005b, Gl. 2)

$$\langle \ddot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle = \ddot{\rho} - \frac{\|\dot{\boldsymbol{X}}_{12}\|^2 - \dot{\rho}^2}{\rho}, \tag{2.7}$$

worin zum einen die ll-SST-Größen (K-Band-Beobachtungen ρ, $\dot{\rho}$, $\ddot{\rho}$) und zum anderen die aus dem hl-SST-System abgeleiteten Positions-, Geschwindigkeits- und Beschleunigungsvektordifferenzen auftreten.

2.2 Schwerefeldbestimmung aus GRACE-Beobachtungen

2.2.1 Beschleunigungsansatz

Mithilfe der Gleichung (2.7) lässt sich unmittelbar eine Beziehung zu Parametern herstellen, die das gesuchte Schwerefeld repräsentieren. Denn die Beschleunigung $\ddot{\boldsymbol{X}}$ des Satelliten entspricht nach der Newton-Eulerschen Bewegungsgleichung

$$\ddot{\boldsymbol{X}} = \nabla V \tag{2.8}$$

dem Gradienten des Gravitationspotentials V, sofern sämtliche nicht-gravitativen Störbeschleunigungen, die Erd- und Meeresgezeiten sowie die Gezeiten anderer Planeten (Sonne, Mond) reduziert sind. Üblicherweise stellt man den Gradienten des Gravitationspotentials in Bezug auf ein lokales Horizontsystem (Index „L“) dar und überführt die Lösung mithilfe zweier Rotationsmatrizen in das raumfeste System gemäß

$$\ddot{\boldsymbol{X}} = \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}} \boldsymbol{R}_{\mathrm{E}}^{\mathrm{L}} \, \nabla_{\mathrm{L}} V(r, \varphi, \lambda). \tag{2.9}$$

Darin bezeichnet $\boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}$ die Rotationsmatrix vom erdfesten ins raumfeste Bezugssystem und $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{L}}$ die Drehmatrix vom lokalen ins erdfeste System. r, φ und λ sind die sphärischen erdfesten Koordinaten des Berechnungspunkts. Abgekürzt gelte

$$\ddot{\boldsymbol{X}} = \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}} \, \nabla_{\mathrm{E}} V(\boldsymbol{x}) \tag{2.10}$$

mit dem kartesischen, erdfesten Positionsvektor $\boldsymbol{x}$ (Austen und Grafarend, 2004, Gl. 4). Aus Gleichung (2.7) ergibt sich somit die Beziehung

$$\langle \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}} \big[\nabla_{\mathrm{E}} V(\boldsymbol{x}_2) - \nabla_{\mathrm{E}} V(\boldsymbol{x}_1) \big], \boldsymbol{e}_{12} \rangle = \ddot{\rho} - \frac{\|\dot{\boldsymbol{X}}_{12}\|^2 - \dot{\rho}^2}{\rho} \tag{2.11}$$

als einfachstes funktionales Modell des Beschleunigungsansatzes (vgl. Austen und Grafarend, 2004, Gl. 7-8). Wegen der Differenz zwischen den *ersten* partiellen Ableitungen des Gravitationspotentials an den *beiden* Satellitenpositionen stellt dieser Zusammenhang ein Zweipunktproblem erster Ordnung dar (Keller und Sharifi, 2005b).

Die unbekannten Stokes-Koeffizienten der Kugelfunktionsentwicklung können nun über eine Ausgleichung nach Gauß-Markov (siehe z.B. Austen und Grafarend, 2004) berechnet werden, wobei die rechte Seite der Gleichung (2.11) den Beobachtungsvektor bildet und der Einheitsvektor $\boldsymbol{e}_{12}$ über Gleichung (2.2) aus den gemessenen Positionen bestimmt werden kann.

2.2.2 Ansatz der Satellitengravitationsgradiometrie

Eine weiterführende, auf dem Beschleunigungsansatz basierende Methode zur Gravitationsfeldbestimmung aus GRACE-Daten interpretiert das GRACE-System als einachsiges Gradiometer mit sehr großer Basislänge (z.B. Hess und Keller, 1999; Sharifi, 2004; Keller und Sharifi, 2005a,b). Die genannten Autoren bedienen sich dabei dem Konzept der Satellitengravitationsgradiometrie (SGG), das auch bei der am 17. März 2009 gestarteten Satellitenmission GOCE Anwendung findet: der Quotient aus gemessener differentieller Beschleunigung (Relativbeschleunigung) und dem Abstand der Prüfmassen approximiert den sog. Schweregradienten (auch Eötvös- bzw. Marussi-Tensor) $\boldsymbol{G}$. Diese Matrix ist mit den zweiten partiellen Ableitungen des Gravitationspotentials V nach dem Ort definiert. Da die Messgröße $\boldsymbol{G}$ (bzw. bei GRACE die Größe $\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}$, s.u.) also direkt proportional zum Potential V an derselben Position $\boldsymbol{X}$ ist, äußert sich eine Krafteinwirkung durch eine Schwerefeldanomalie instantan in der gemessenen Relativbeschleunigung beider Prüfmassen (Hess und Keller, 1999), und zwar – anders als bei der klassischen Analyse von Satellitenbahnstörungen mittels hl-SST (z.B. nach Colombo, 1984) – unabhängig von früher einwirkenden Schwerefeldanomalien.

Neben der direkten Beziehung zur Gravitationsfeldgeometrie bietet das SGG-Konzept einen weiteren Vorteil: Da sich die Beobachtungsgröße $\boldsymbol{G}$ auf die zweiten anstatt auf die ersten Ableitungen des Gravitationspotentials bezieht, werden höherfrequente Signalanteile verstärkt. Das Rauschen wird zwar auch verstärkt, hat jedoch ein anderes spektrales Verhalten, sodass das SGG-Verfahren letztlich zu einer verbesserten Auflösung führt. Im Falle von GOCE erhofft man sich, mithilfe des Gradiometrie-Ansatzes das Gravitationsfeld bis in sehr kurzwellige Bereiche von $n > 200$ auflösen zu können.

Für die GRACE-Mission entwickelten u.a. Keller und Sharifi Beziehungen zwischen den GRACE-Beobachtungen und der Gradiometrie-Messgröße, dem Schweregradienten $\boldsymbol{G}$. Zufolge Keller und Sharifi (2005b) gilt für die Differenz der Gravitationsbeschleunigungen in linearer Näherung

$$\begin{aligned} \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}\big[\nabla_{\mathrm{E}}V(\boldsymbol{x}_2) - \nabla_{\mathrm{E}}V(\boldsymbol{x}_1)\big] &\approx \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}\big[\nabla_{\mathrm{E}}\langle \boldsymbol{x}_{12}, \frac{\partial V}{\partial \bar{\boldsymbol{x}}_{12}}\rangle\big] \\ &\approx \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}\big[\nabla^{\top} \otimes \nabla V(\bar{\boldsymbol{x}}_{12})\big] \cdot \boldsymbol{X}_{12} \\ &\approx \boldsymbol{G} \cdot \boldsymbol{X}_{12}, \end{aligned} \tag{2.12}$$

wobei $\boldsymbol{G}$ der unbekannte Schweregradient, $\bar{\boldsymbol{x}}_{12}$ der geometrische Mittelpunkt der beiden Satelliten[1] und $\otimes$ das Kronecker-Produkt bezeichnet. Mit Gl. (2.11) folgt daraus die sogenannte *lineare Gradiometriegleichung*

$$\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{X}_{12} \approx \ddot{\rho} - \frac{\|\dot{\boldsymbol{X}}_{12}\|^2 - \dot{\rho}^2}{\rho} \quad \overset{\cdot\rho^{-1}}{\Longrightarrow} \quad \boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12} \approx \frac{\ddot{\rho}}{\rho} + \frac{\dot{\rho}^2}{\rho^2} - \frac{\|\dot{\boldsymbol{X}}_{12}\|^2}{\rho^2}. \tag{2.13}$$

Mathematisch betrachtet wird das oben genannte Zweipunktproblem des Beschleunigungsansatzes hiermit in ein Einpunktfunktional höherer (mindestens zweiter) Ordnung überführt, da Gl. (2.13) die *zweiten* Ableitungen des Gravitationspotentials am *Baryzentrum* des Satellitensystems (linke Seite der Gleichung) mit den Beobachtungsgrößen (rechte Seite) verknüpft (Keller und Sharifi, 2005b).

Indem man den in LOS-Richtung projizierten Eötvös-Tensor $\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}$ als Linearkombination der unbekannten Schwerefeldkoeffizienten schreibt, erhält man eine in ebendiesen Unbekannten lineare Beobachtungsgleichung. Es ist somit wiederum eine Ausgleichung möglich, wobei die rechte Seite der Gl. (2.13) den Beobachtungsvektor bildet und die Zeilenanzahl der Designmatrix der Anzahl der Beobachtungszeitpunkte entspricht (Keller und Sharifi, 2005b). Die Simulationsstudie von Hess und Keller (1999) zeigt allerdings auf, dass der Fehlereinfluss, der durch den Abbruch der Taylorreihenentwicklung nach der ersten Ordnung entsteht, im Falle von GRACE aufgrund des großen Prüfmassenabstandes ρ nicht vernachlässigbar ist (die Abweichung betrage bzgl. $\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}$ etwa $0.55\,\mathrm{E}^*$).

Um dem Linearisierungsfehler entgegenzuwirken, kann man zufolge Keller und Sharifi (2005b) entweder höhere Entwicklungsgrade berücksichtigen oder den dort beschriebenen Korrekturterm an die lineare Gradiometriegleichung anbringen, der den Einfluss niedriger sphärisch-harmonischer Grade auf die Beobachtungsgrößen mittels eines a priori Referenzfelds eliminiert. Mit beiden Maßnahmen ließe sich der Abbruchfehler auf wenige Milli-Eötvös reduzieren.

Allerdings ist zu beachten, dass bei GRACE keineswegs alle Tensorkomponenten mit dieser Genauigkeit erfassen werden können, sondern hauptsächlich nur die x-Komponente (V_{xx}), denn die Richtung der Verbindungslinie, auf die die Größe $\boldsymbol{G}$ bezogen wird, entspricht ungefähr der along-track-Richtung des Hill-Dreibeins. Dies ist eine konzeptionelle Schwäche der GRACE-Messanordnung an sich und zeigt sich bei allen anderen Ansätzen zur Schwerefeldbestimmung aus GRACE-Beobachtungen in analoger Weise. In Kombination mit der polnahen Umlaufbahn führt die Messanordnung zu den typischen, unrealistischen Streifen in Nord-Süd-Richtung, die man oftmals in GRACE-Schwerefeldmodellen erkennen kann und erst dann weitgehend verschwinden, sobald z.B. terrestrische Messungen in die Schwerefeldbestimmung miteinfließen[2]. Um die Tensorkomponenten unabhängig voneinander zu bestimmen, wären andere Satellitenanordnungen, die für das lokale Gravitationsfeld in mehreren Raumachsen sensitiv sind, besser geeignet (vgl. Sneeuw und Schaub, 2005; Sneeuw et al., 2008). Diesbezüglich ist GOCE, das mit seinen drei orthogonalen „Messach-

1 Da die Satelliten in etwa die gleiche Masse haben, entspricht der Mittelpunkt auch dem Baryzentrum
* Eötvös-Einheit: $1\,\mathrm{E} \mathrel{\widehat{=}} 10^{-9}\,\mathrm{s}^{-2}$
2 Siehe z.B. http://op.gfz-potsdam.de/grace/results/grav/g007_eigen-05c.html

sen“ quasi als Volltensor-Gradiometer gelten kann, der GRACE-Messanordnung überlegen. Laut Rummel (2003) ist allerdings die Genauigkeit des fiktiven GRACE-Gradiometers in LOS-Richtung mit $10^{-6}\,\mathrm{E}/\sqrt{\mathrm{Hz}}$ aufgrund seiner langen Basis besser als die along-track-Komponente bei GOCE ($4 \cdot 10^{-6}\,\mathrm{E}/\sqrt{\mathrm{Hz}}$). Insofern scheint der Ansatz der Satellitengravitationsgradiometrie für GRACE durchaus vielversprechend zu sein.

Hat man das Ziel, das theoretisch hohe Genauigkeitspotential des einachsigen „GRACE-Gradiometers“ auch in der Praxis voll auszuschöpfen, stellt die geeignete Kombination der unterschiedlichen GRACE-Beobachtungen die wohl größte Limitierung dar. Denn wie anhand Gleichung (2.13) deutlich wird, gehen zum einen die weniger genauen (und zeitlich weniger häufig diskretisierten) GPS-Messungen und zum anderen die hochgenauen (und dicht diskretisierten) K-Band-Beobachtungen in den Beobachtungsvektor ein. Diese Problematik tritt auch bei dem angesprochenen Beschleunigungsansatz auf (vgl. Gl. (2.11)) und ebenso bei anderen Ansätzen, wie etwa bei dem in Novák et al. (2006) vorgestellten *spacewise approach*, wobei die Differenz $\nabla V(\boldsymbol{x}_2) - \nabla V(\boldsymbol{x}_1)$ durch ein Abel-Poisson-Randwertintegral ersetzt wird, das man anschließend bezüglich einer Referenz-Randwertfläche lösen kann.

Im folgenden Abschnitt soll daher exemplarisch am Gradiometrieansatz untersucht werden, welche Genauigkeitsanforderungen an die beobachteten Größen zu stellen sind, um eine bestimmte Genauigkeit für das Erdschwerefeld bzw. den Schweregradienten zu gewährleisten.

2.2.3 Fehlerabschätzung für die SGG-Auswertung

Zur Abschätzung der Fehler genüge hier die einfache lineare Gradiometriegleichung

$$\boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12} = \frac{\ddot{\rho}}{\rho} + \frac{\dot{\rho}^2}{\rho^2} - \frac{\|\dot{\boldsymbol{X}}_{12}\|^2}{\rho^2}$$

entsprechend Gl. (2.13). Wertet man diese nach der linearen Fehlerfortpflanzung gemäß

$$\Delta \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12} = \left|\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \rho}\right| \Delta\rho + \left|\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \dot{\rho}}\right| \Delta\dot{\rho} + \left|\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \ddot{\rho}}\right| \Delta\ddot{\rho} + \left|\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \|\dot{\boldsymbol{X}}_{12}\|^2}\right| \Delta\|\dot{\boldsymbol{X}}_{12}\|^2 \quad (2.14)$$

aus, folgt aus den Beobachtungsfehlern näherungsweise der größtmögliche Fehler des Schweregradienten. Die dabei benötigten partiellen Ableitungen nach den Beobachtungen lauten

$$\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \rho} = \frac{\ddot{\rho}}{\rho^2} + \frac{2\dot{\rho}^2}{\rho^3} - \frac{2\|\dot{\boldsymbol{X}}_{12}\|^2}{\rho^3} \quad (2.15)$$

$$\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \dot{\rho}} = \frac{2\dot{\rho}}{\rho^2} \quad (2.16)$$

$$\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \ddot{\rho}} = \frac{1}{\rho} \quad (2.17)$$

$$\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \|\dot{\boldsymbol{X}}_{12}\|^2} = -\frac{1}{\rho^2}. \quad (2.18)$$

In Gleichung (2.14) sind jeweils die Beträge der Ableitungen einzusetzen, damit sich die Einzelfehler nicht gegenseitig kompensieren. Da man den ungünstigsten Fall für $\Delta \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}$ annehmen muss (Maximalfehler), sind zudem bei den Ableitungen die jeweils größtmöglichen Beträge anzusetzen. Daher wird anstelle der ursprünglichen Ableitung in (2.15) deren obere Grenze angenommen:

$$\frac{\partial \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12}}{\partial \rho} \leq \sqrt{\left(\frac{\ddot{\rho}}{\rho^2}\right)^2 + \left(\frac{2\dot{\rho}^2}{\rho^3}\right)^2 + \left(\frac{2\|\dot{\boldsymbol{X}}_{12}\|^2}{\rho^3}\right)^2}. \quad (2.19)$$

Zur Fehlerabschätzung sind Zahlenwerte für die Beobachtungsgrößen einer typischen GRACE-Konstellation erforderlich. Hier wurden aus den zur Verfügung gestellten Daten[1] folgende Werte ermittelt, die realistisch sind aber zugleich verhältnismäßig große partielle Ableitungen ergeben (Minimalwert für ρ, im Übrigen jeweils Maximalwerte):

$$\left.\begin{array}{ll} \rho = 219289.879\,\text{m}, & \dot{\rho} = 2.2937\,{}^{\text{m}}/_{\text{s}} \\ \ddot{\rho} = 0.0021\,{}^{\text{m}}/_{\text{s}^2}, & \|\dot{\boldsymbol{X}}_{12}\| = 249.3021\,{}^{\text{m}}/_{\text{s}} \end{array}\right\} \xrightarrow{(2.13)} \boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12} = -1282.7627\,\text{E}$$

Stellt man nun die Fehleranteile einzeln der Genauigkeit $\Delta\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}$ gegenüber, z.B.

$$\Delta\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12} = \left|\frac{\partial\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}}{\partial\rho}\right|\Delta\rho \iff \frac{\Delta\rho}{\rho} = \left(\left|\frac{\partial\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}}{\partial\rho}\right|\right)^{-1}\frac{\Delta\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12}}{\rho} \tag{2.20}$$

so resultieren die in Abbildung 2.3 dargestellten relativen Fehler der Beobachtungen. Die gemessenen Relativdistanzen ρ müssen demnach etwa auf sieben signifikante Stellen genau bekannt sein, damit der Schweregradient mit einer Genauigkeit von einem Milli-Eötvös bestimmt werden kann.

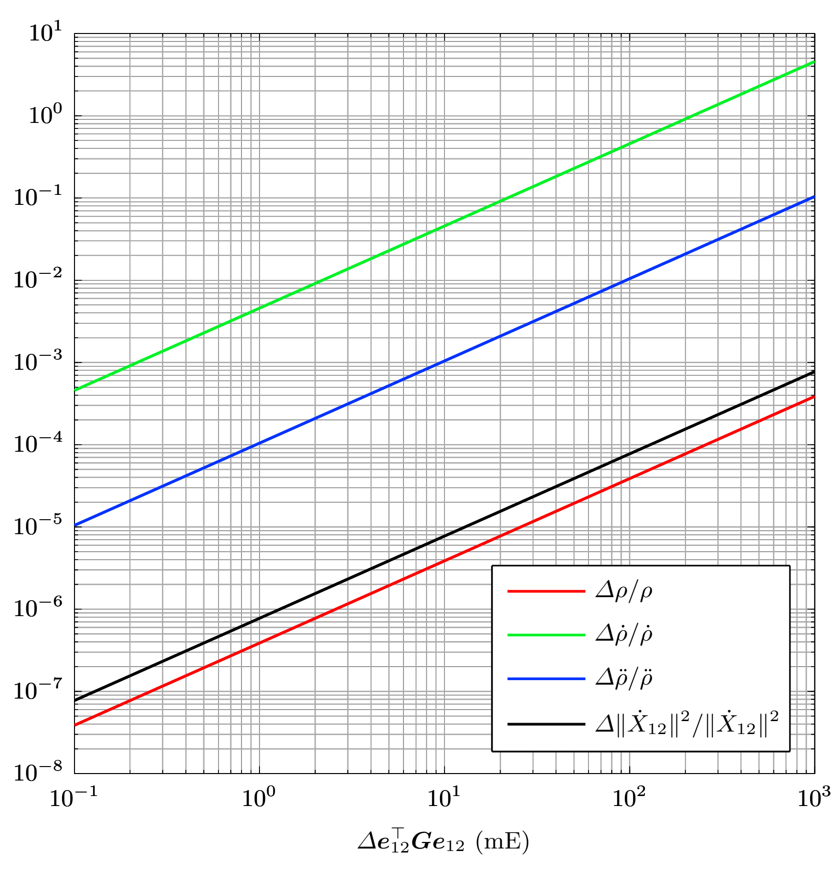


Abb. 2.3: Fehlerabschätzung: relative Fehler vs. Schweregradienten-Genauigkeit

Der maximale Absolutfehler der Relativdistanz darf höchstens etwa acht Zentimeter betragen, sofern man wiederum die absolute Genauigkeit des Schweregradienten von $\Delta\boldsymbol{e}_{12}^{\top}\boldsymbol{G}\boldsymbol{e}_{12} = 1\,\text{mE}$ fordert und nur den ersten Term auf der rechten Seite der Gl. (2.14) berücksichtigt. Analysen haben gezeigt, dass mit dem K-Band-Messsystem zuverlässig Standardabweichungen von $\sigma_\rho < 10\,\mu\text{m}$ und $\sigma_{\dot{\rho}} < 1\,{}^{\mu\text{m}}/_{\text{s}}$ erreicht werden (vgl. Dunn et al., 2002). Anforderungen in der Größenordnung von einigen Zentimetern sind somit unproblematisch. Bei der Relativbeschleunigung $\ddot{\rho}$ handelt es sich (wie auch bei $\dot{\rho}$) nicht um einen unmittelbaren Messwert des K-Band-Instruments. Stattdessen wird diese „Pseudobeobachtung" bei der Level-1B-Prozessierung durch numerische Doppeldifferentiation

1 Die Daten umfassen sämtliche GPS-Navigationslösungen und K-Band-Beobachtungen des 01.01.2003. Nähere Informationen zu den realen GRACE-Daten siehe Abschnitt 4.1.1 und Abschnitt 5.1.

aus der eigentlichen Messung ρ berechnet, d.h. die Standardabweichung der Relativgeschwindigkeit und -beschleunigung hängt von der Messgenauigkeit der Relativentfernung ρ ab. Demzufolge kann man mithilfe des zentralen Differenzenquotienten

$$f'(x_0) \approx f_z(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} \tag{2.21}$$

und Anwendung des Fehlerfortpflanzungsgesetzes

$$\sigma^2_{f_z(x_0)} = \left(\frac{1}{2h}\right)^2 \sigma^2_{f(x_0+h)} + \left(\frac{1}{2h}\right)^2 \sigma^2_{f(x_0-h)} \tag{2.22}$$

die Standardabweichung $\sigma_{\ddot{\rho}} = 0.1414\,\mathrm{\mu m/s^2}$ für die Relativbeschleunigung abschätzen. In ihrer Simulation verwenden Keller und Sharifi (2005b) den ähnlichen Wert $0.1398\,\mathrm{\mu m/s^2}$ für die zufällige Abweichung von $\ddot{\rho}$.

Tabelle 2.1 listet die zeitlichen Auflösungen und die zu erwartenden Standardabweichungen der GRACE-Daten auf. Im Datensatz der GPS-Navigationslösung sind bereits formale Fehler der Positions- und Geschwindigkeitskomponenten enthalten. Die hier angegebenen Werte wurden aus den Daten des GRACE-B Satelliten vom 1. Januar 2003 extrahiert, indem jeweils Maximum, Minimum und Mittelwert (in Klammern angegeben) aus allen gegebenen Standardabweichungen berechnet wurden. Es handelt sich hierbei also nicht um den gesamten Punktfehler, sondern die Angaben beziehen sich jeweils auf eine einzelne Komponente (x, y, z). Da nach Jekeli et al. (1999) die Standardabweichungen der Geschwindigkeitskomponenten kinematischer Orbits typischerweise um $0.1\,\mathrm{mm/s}$ betragen, erscheinen die im Datensatz GNV1B verzeichneten Geschwindigkeitsfehler zunächst recht optimistisch. Die GNV1B-Orbits sind jedoch reduziert-dynamische, und die Größenordnung der angegebenen Fehler bestätigt sich später beim Vergleich mit den hochgenauen K-Band-Messungen.

Zu den K-Band-Beobachtungen liegen im zugehörigen Datensatz (KBR1B) selbst keine Qualitätsinformationen vor. In der Tabelle sind daher nochmals die bereits erwähnten Genauigkeitsangaben aus Dunn et al. (2002) und die daraus abgeleitete Standardabweichung $\sigma_{\ddot{\rho}}$ aufgeführt.

Tab. 2.1: Spezifikationen der GRACE-Datensätze

Datensatz	Messwert	Standardabweichung	Abtastintervall
GPS-Navigations-lösung (GNV1B)	$\boldsymbol{X}_1, \boldsymbol{X}_2$ $\dot{\boldsymbol{X}}_1, \dot{\boldsymbol{X}}_2$	$1 - 5\,\mathrm{cm}$ ($2\,\mathrm{cm}$) $0.01 - 0.05\,\mathrm{mm/s}$ ($0.02\,\mathrm{mm/s}$)	60 Sek.
K-Band-Messung (KBR1B)	ρ $\dot{\rho}$ $\ddot{\rho}$	$< 10\,\mathrm{\mu m}$ $< 1\,\mathrm{\mu m/s}$ $\approx 0.14\,\mathrm{\mu m/s^2}$	5 Sek.

Mit ebendiesen Genauigkeiten für die K-Band-Beobachtungen und der geforderten Genauigkeit $\Delta \boldsymbol{e}_{12}^\top \boldsymbol{G} \boldsymbol{e}_{12} = 1\,\mathrm{mE}$ folgt nun aus Gleichung (2.14) und unter Berücksichtigung des Zusammenhangs

$$\Delta\|\dot{\boldsymbol{X}}_{12}\|^2 = \left|\frac{\partial\|\dot{\boldsymbol{X}}_{12}\|^2}{\partial\|\dot{\boldsymbol{X}}_{12}\|}\right| \Delta\|\dot{\boldsymbol{X}}_{12}\| = 2\|\dot{\boldsymbol{X}}_{12}\| \cdot \Delta\|\dot{\boldsymbol{X}}_{12}\|, \tag{2.23}$$

dass der Absolutfehler der Relativgeschwindigkeit $\Delta\|\dot{\boldsymbol{X}}_{12}\|$ höchstens $34.2\,\mathrm{\mu m/s}$ betragen darf. In einer ähnlichen Abschätzung (Weigelt und Keller, 2008) werden hierfür sogar $5\,\mathrm{\mu m/s}$ beansprucht.

Vergleicht man diese Anforderung mit den Genauigkeiten der realen GPS-Navigationslösung (Tab. 2.1), erkennt man, dass die aus der Navigationslösung abgeleitete Geschwindigkeitsdifferenz

$\|\dot{\boldsymbol{X}}_{12}\|$ tatsächlich die kritischste Größe für die Qualität der Gravitationsfeldlösung darstellt.

Da der Differenzvektor $\dot{\boldsymbol{X}}_{12}$, wie oben anhand Gleichung (2.3) gezeigt wurde, mit der K-Band-Relativgeschwindigkeit $\dot{\rho}$ korreliert ist, macht es also durchaus Sinn, die hochgenauen Messungen $\dot{\rho}$ als Zwangsbedingung bei der Orbitverdichtung zu nutzen, um dadurch die Genauigkeit der Geschwindigkeitsdifferenz $\|\dot{\boldsymbol{X}}_{12}\|$ möglichst zu verbessern.

2.3 Untersuchung einfacher Verfahren zur Orbitverdichtung

Im nachfolgenden Abschnitt werden einige einfache Methoden zur Verdichtung von GRACE-Orbits untersucht, um aufzuzeigen, dass diese nicht den hohen Genauigkeitsansprüchen genügen[1].

Dieser Nachweis wird in MATLAB anhand simulierter Daten erbracht. Wie man die Daten erstellt, sei an dieser Stelle nur kurz skizziert[2]: zunächst berechnet man mithilfe des in dieser Arbeit umgesetzten Orbitintegrators unter Berücksichtigung des Gravitationsfeldmodells GGM02s bis Grad/Ordnung 20 für beide Satelliten Referenzorbits – genauer gesagt Positions- und Geschwindigkeitszeitreihen über 10 Minuten bezüglich des raumfesten Bezugssystems –, ausgehend jeweils von einem sechsdimensionalen Bahnvektor, der drei Positions- und drei Geschwindigkeitskomponenten enthält. Letztgenannte Anfangsbedingungen werden aus den zur Verfügung gestellten realen GRACE-Daten vom 01.01.2003 entnommen. Im Anschluss leitet man simulierte Navigationslösungen her, indem man das Sampling der Referenzorbits auf 60 Sekunden reduziert und die Daten mit normalverteiltem, mittelwertfreiem Rauschen überlagert (Standardabweichungen von 2 cm bzgl. der Positions- und 0.02 mm/s bzgl. der Geschwindigkeitskomponenten).

Die so erzeugten, verrauschten Ephemeriden werden zuerst mit reinen Interpolationsverfahren und anschließend mit Integrationsverfahren verdichtet und jeweils mit den Referenzorbits verglichen.

2.3.1 Lineare Interpolation

Das wohl einfachste Interpolationsverfahren ist die lineare Interpolation mithilfe der bekannten Lagrange- bzw. Newton-Formel ersten Grades (s. Tafel 2.1).

Tafel 2.1: Lineare Interpolation*

Gegeben sei eine reelle Funktion $f : I \to \mathbb{R}$ auf dem Grundintervall $I = [a, b] \subset \mathbb{R}$, wobei $-\infty < a < b < \infty$. Mit den Funktionswerten $f_0 = f(t_0)$, $f_1 = f(t_1)$ an den Stützstellen $a \leq t_0 < t_1 \leq b$ gilt dann für den linear interpolierten Wert an der Stelle t_i:

$$f_{\text{lin.}}(t_i) = f_0 + \frac{f_1 - f_0}{t_1 - t_0}(t_i - t_0) \tag{2.24}$$

* Vgl. HOFFMANN ET AL. (2005, S. 798 u. 805)

In MATLAB steht der Befehl `interp1` zur Verfügung, um die eindimensionale Funktion f linear zu interpolieren. Geometrisch betrachtet werden die Bahnpunkte hierbei durch Raumgeraden verbunden. Da die Satellitenbahn jedoch in erster Näherung ellipsenförmig verläuft, ist die Abweichung zum Referenzorbit etwa in der Mitte zwischen zwei Bahnpunkten maximal (vgl. Abb. 2.4).

1 Der geneigte Leser kann die nachfolgend beschriebenen Verfahren und Ergebnisse anhand der MATLAB-Datei `simple_odm_sim_main` im Verzeichnis `\ekf_orbit_toolbox` der CD-ROM nachvollziehen.

2 Die Vorgehensweise ist hier analog zu den in Abschnitt 4.7 ausführlich beschriebenen Schritten.

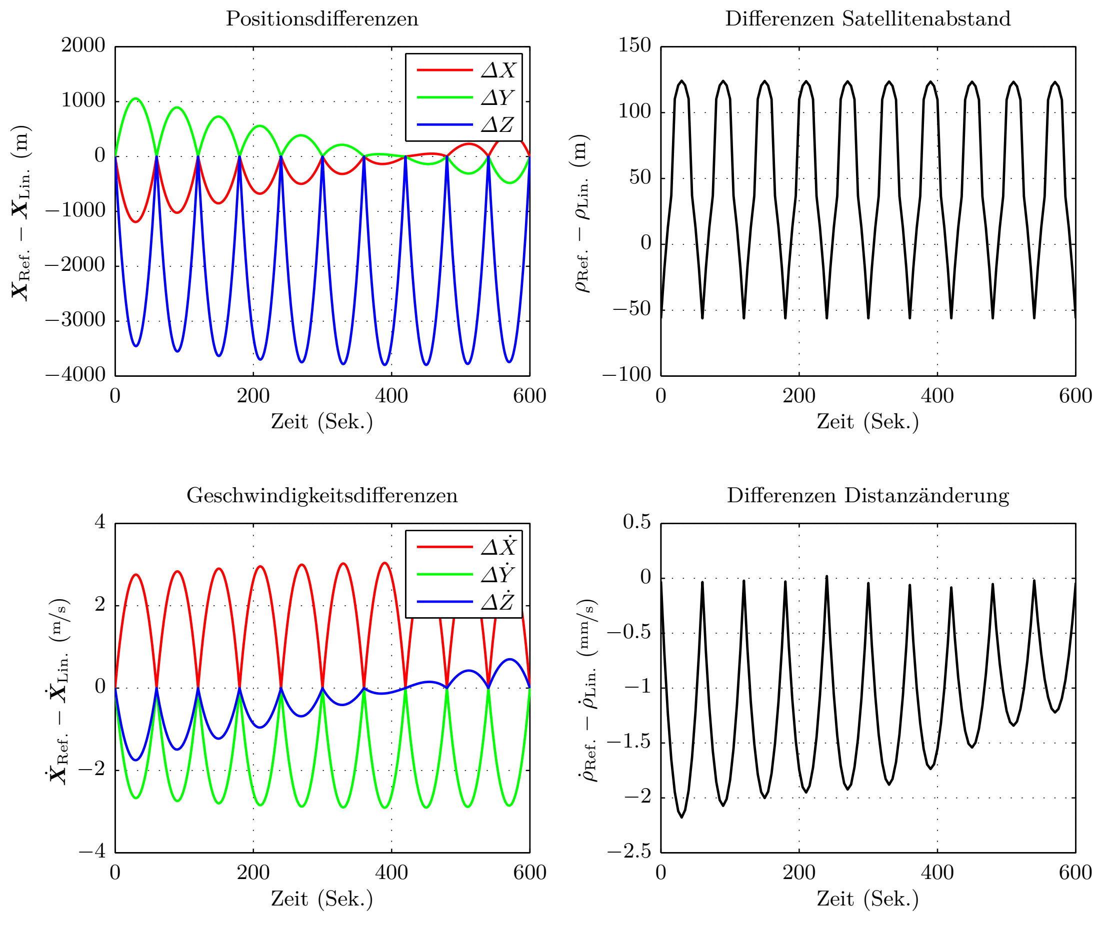


Abb. 2.4: Referenz vs. lineare Verdichtung

Aus Platzgründen sind nur die Positions- und Geschwindigkeitsdifferenzen des ersten Satelliten dargestellt. Für den zweiten Satelliten erhält man ähnlich hohe Abweichungen bis fast 4 km bezüglich der Positions- und bis ca. 3 m/s bezüglich der Geschwindigkeitskomponenten. Mithilfe der Gleichungen (2.1) bis (2.3) lassen sich außerdem die Relativentfernungen $\rho_{\text{Lin.}}$ und -geschwindigkeiten $\dot{\rho}_{\text{Lin.}}$ aus den linear verdichteten Satellitenpositionen und -geschwindigkeiten berechnen (mithilfe der dazu umgesetzten MATLAB-Funktion `grace_parms`, s. Anhang A.2, S. 138) und den entsprechenden Referenzwerten gegenüberstellen (Abb. 2.4 rechts).

2.3.2 Kubische Spline-Interpolation

Die Ergebnisse der linearen Interpolation sind ganz offensichtlich nicht zufriedenstellend. Darum wurde als Nächstes die kubische Spline-Interpolation getestet. Nähere mathematische Betrachtungen zu beiden Interpolationstechniken sind beispielsweise in HOFFMANN ET AL. (2005, S. 805 ff.) zu finden. Die Wahl fiel hier auf kubische Splines, da sie im Vergleich zu herkömmlichen Interpolationspolynomen meist deutlich weniger Oszillationen aufweisen.

Tafel 2.2: Kubische Spline-Interpolation*

Die zu interpolierende Funktion habe an $n+1$ Knoten $\{a = t_0 < t_1 < \ldots < t_n = b\}$ ebenso viele Stützwerte $y_j = f(t_j),\ j = 0, \ldots, n \in \mathbb{N}$.

$s : [a, b] \to \mathbb{R}$ heißt kubischer Spline, falls

- s auf $[a, b]$ zweimal stetig differenzierbar ist, d.h. $s \in C^2([a, b])$, und
- s sich stückweise aus n kubischen Polynomen p_j zusammensetzt:

$$s(t)|_{[t_j, t_{j+1}]} \equiv p_j(t) \text{ mit} \tag{2.25}$$

$$p_j(t) = \alpha_j + \beta_j(t - t_j) + \delta_j(t - t_j)^2 + \gamma_j(t - t_j)^3 \quad \forall\, j \in [0, n-1] \tag{2.26}$$

* Vgl. Freund und Hoppe (2007, Def. 2.4.1.1); Iske (2007, S. 66)

Bei der Interpolation mittels kubischer Splines sind also vier unbekannte Polynomkoeffizienten $(\alpha, \ldots, \gamma)$ in jedem Teilintervall $[t_j, t_{j+1}]$ zu bestimmen. Dazu werden neben der herkömmlichen Interpolationsbedingung (Splinefunktion verläuft durch Stützpunkte der Originalfunktion, d.h. $s(t_j) = f(t_j)$) weitere Anschluss- und Stetigkeitsbedingungen an den Übergangsstellen der einzelnen Teilpolynome aufgestellt. Für die Festlegung der zwei letzten Freiheitsgrade gibt es unterschiedliche Ansätze. Die Matlab-eigene Spline-Interpolation, die sich u.a. gemäß der Syntax `yi = spline(t,y,ti)` starten lässt, nutzt hierzu die sog. „not-a-knot"-Bedingung. Diese fordert Stetigkeit bezüglich der dritten Ableitungen $p_0'''(t_1) = p_1'''(t_1)$ bzw. $p_{n-2}'''(t_{n-1}) = p_{n-1}'''(t_{n-1})$, sodass die Splinefunktion $s(x)$ auf den Intervallen $[t_0, t_1]$ und $[t_1, t_2]$ bzw. $[t_{n-2}, t_{n-1}]$ und $[t_{n-1}, t_n]$ identisch ist. Es werden also jeweils die ursprünglichen zwei Polynome von zwei Teilintervallen zu einem Polynom zusammengefasst, wodurch die zweite und vorletzte Stützstelle (t_1 und t_{n-1}) im eigentlichen Sinn „keine Knoten" mehr sind. Beim Vergleich mit den Referenzorbits macht sich dieser Umstand durch größere Abweichungen am Anfang und Ende des 10-Minuten-Bogens bemerkbar (vgl. Abb. 2.5), was als nachteilig anzusehen ist.

Von Vorteil ist allerdings, dass sich die Bedingungsgleichungen zur Bestimmung der Polynomkoeffizienten in Form eines tridiagonalen linearen Gleichungssystems aufstellen lassen. Ein solches System lässt sich bekanntlich gerade in Matlab effizient auflösen.

Im Vergleich zur linearen Interpolation ist die kubische Spline-Interpolation zwar schon wesentlich genauer, genügt aber ebenfalls nicht den Anforderungen. Wie eingangs erwähnt, sind für die Geschwindigkeitsdifferenz $\Delta\|\dot{\boldsymbol{X}}_{12}\|$ Genauigkeiten in der Größenordnung von wenigen $\mu\mathrm{m/s}$ wünschenswert. Wegen der Beziehung (2.3) sollte ein vergleichbar hoher Anspruch auch für die aus den verdichteten Orbits bestimmte Relativgeschwindigkeit (hier $\dot{\rho}_{\text{spl.}}$) gelten. Daher kann man die Abweichungen dieser Werte zur Referenz $\dot{\rho}_{\text{Ref.}}$ als Indiz dafür heranziehen, ob der verdichtete Orbit die angestrebte Genauigkeit für $\Delta\|\dot{\boldsymbol{X}}_{12}\|$ erfüllt. Hierzu müssten die Differenzen $\dot{\rho}_{\text{Ref.}} - \dot{\rho}_{\text{spl.}}$ in Abb. 2.5 wenigstens in einer Größenordnung um 10^{-6} m/s liegen.

2.3.3 Verdichtung mittels ungestörter Keplerbahnen

Während die bisher betrachteten reinen Interpolationsverfahren mathematisch begründet sind, steht mit den Methoden der Orbitintegration ein Werkzeug zur Verfügung, das dem physikalischen Hintergrund der Satellitenbewegung eher gerecht wird. Als erster Versuch bietet sich eine einfache Bahnintegration für den ungestörten Keplerfall an. Dabei betrachtet man die Newtonsche Bewegungsgleichung mit dem ersten Bahnpunkt $\boldsymbol{X}(t_0), \dot{\boldsymbol{X}}(t_0)$ als das Anfangswertproblem

$$\ddot{\boldsymbol{X}}(t) = -\frac{GM}{\|\boldsymbol{X}(t)\|^3} \cdot \boldsymbol{X}(t), \quad \boldsymbol{X}(t_0) = \boldsymbol{X}_0, \quad \dot{\boldsymbol{X}}(t_0) = \dot{\boldsymbol{X}}_0, \tag{2.27}$$

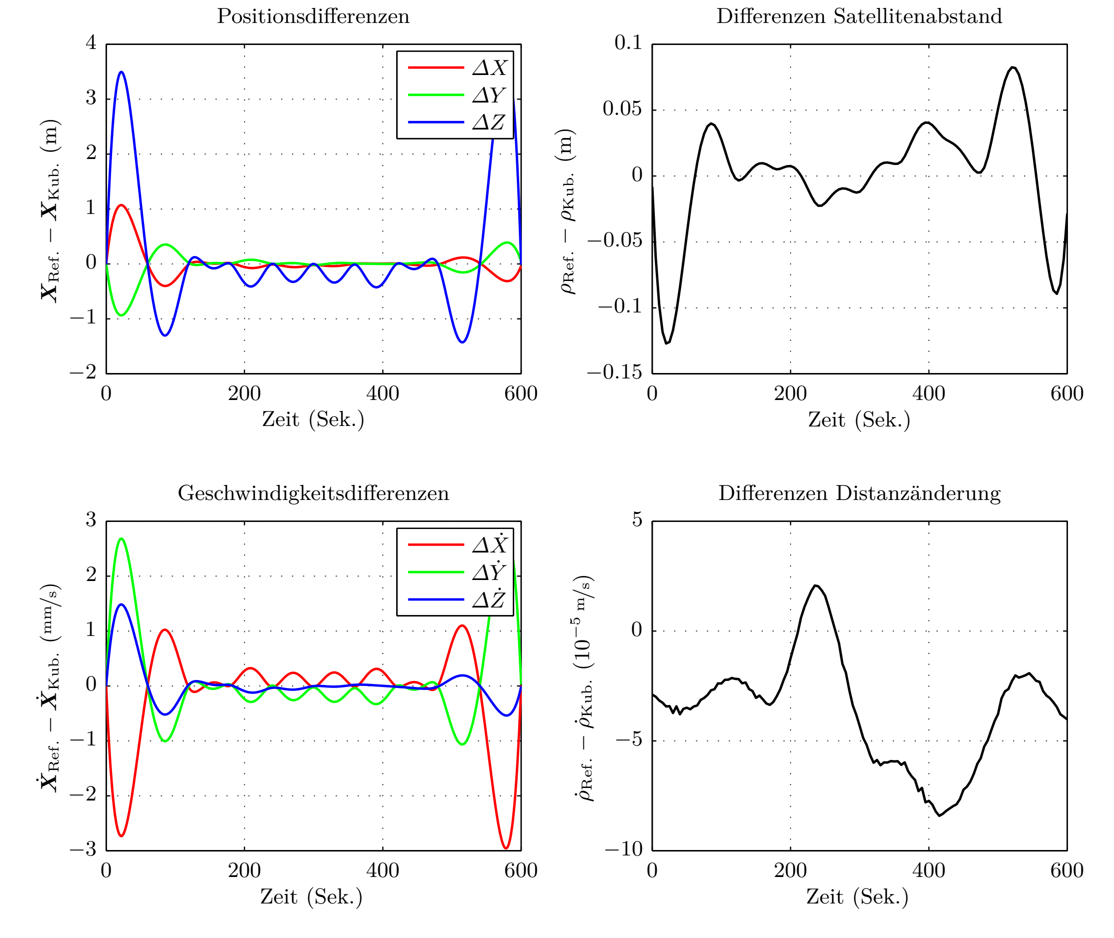


Abb. 2.5: Referenz vs. kubischer Spline-Verdichtung

wobei $GM = \text{const.} \overset{\text{def}}{=} 3.986004415 \cdot 10^{14}\ {}^{\text{m}^3}/_{\text{s}^2}$ die geozentrische Gravitationskonstante bezeichnet[1]. Bei Kenntnis der sechs Integrationskonstanten

$$X_0,\ Y_0,\ Z_0,\ \dot{X}_0,\ \dot{Y}_0,\ \dot{Z}_0$$

zu einem Bezugszeitpunkt t_0 kann die Bahn des Satelliten z.B. durch numerische Integration der obigen drei gewöhnlichen Differentialgleichungen (DGL) zweiter Ordnung vollständig bestimmt werden. Äquivalente Konstanten sind die bekannten sechs Keplerschen Bahnelemente (z.B. Seeber, 1989, S. 64):

- a Große Halbachse
- e Erste numerische Exzentrizität
- i Inklination
- Ω Rektaszension des aufsteigenden Knotens
- ω Argument des Perigäums
- ν Wahre Anomalie

Um die Position des Satelliten in der Bahnebene festzulegen, ist anstelle der wahren Anomalie auch die exzentrische Anomalie E sowie die mittlere Anomalie M gebräuchlich.

Die Keplerelemente ermöglichen durch die sog. Ephemeridenrechnung (vgl. Seeber, 1989, S. 112 ff.) eine alternative Lösung der vektoriellen DGL (2.27), d.h. ausgehend vom ersten Bahnpunkt lassen sich zukünftige Positionen und Geschwindigkeiten des Satelliten ermitteln. Der Ablauf der Ephemeridenrechnung ist in Tafel 2.3 skizziert. Im Wesentlichen berechnet man in jedem Bahnpunkt des gegebenen unverdichteten Orbits eine Keplerellipse und setzt diese bis zum nächsten Bahnpunkt

1 Def. gemäß den meisten Gravitationsfeldmodellen, s. z.B. Datei `constants_egm08` im Verzeichnis `\gravfields`.

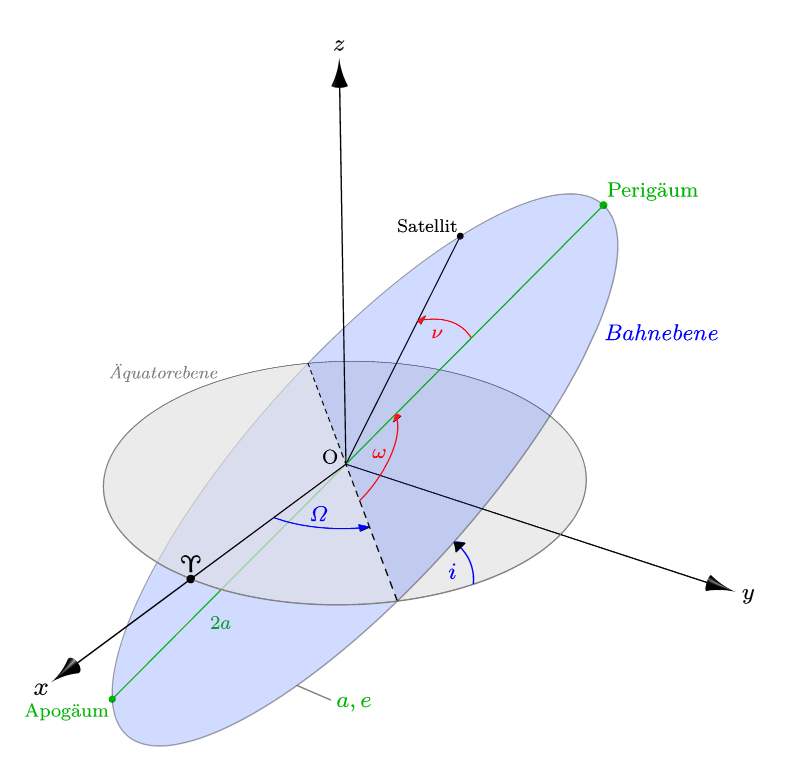


Abb. 2.6: Keplersche Bahnparameter

fort. Die notwendigen Transformationen zwischen den Positionen $\boldsymbol{X}$ und Geschwindigkeiten $\dot{\boldsymbol{X}}$ im raumfesten Äquatorsystem und den zugehörigen Keplerelementen (`kepler2cart`, `cart2kepler`) sind Standardalgorithmen der Satellitengeodäsie und werden hier als bekannt vorausgesetzt (siehe ggf. z.B. Kaula, 1966, Abschn. 3.1; Seeber, 1989, S. 113; Vallado und McClain, 2007, Abschn. 2.5 u. 2.6; Curtis, 2010, Alg. 4.2 bzw. 4.5).

Tafel 2.3: Ephemeridenrechnung

- **Schritt 1:** Berechne die Keplerelemente an allen Bahnpunkten des unverdichteten Orbits:

$$\boldsymbol{X}_0, \dot{\boldsymbol{X}}_0 \xrightarrow{\texttt{cart2kepler}} \{a, e, i, \omega, \Omega, M_0\}$$

- **Schritt 2:** Prädiziere die mittlere Anomalie M (mit $\Delta t = 5\,\mathrm{s}$) bis zum jeweils nächsten Bahnpunkt:

$$M(t_i) = M_0 + n \cdot (t_i - t_0), \quad t_i = t_0 + i \cdot \Delta t.$$

  Die mittlere Bewegung n ergibt sich dabei aus dem dritten Keplerschen Gesetz

$$n = \sqrt{GM/a^3}.$$

- **Schritt 3:** Bestimme aus den nun alle fünf Sekunden vorliegenden Keplerelementen (dabei sind $\{a, e, i, \omega, \Omega\}$ = const. im entspr. 60-Sek.-Bogen) jeweils den Orts- und Geschwindigkeitsvektor im quasi-inertialen Bezugssystem:

$$\{a, e, i, \omega, \Omega, M(t_i)\} \xrightarrow{\texttt{kepler2cart}} \boldsymbol{X}(t_i), \dot{\boldsymbol{X}}(t_i)$$

Die Resultate der Orbitverdichtung mittels der Ephemeridenrechnung (Abb. 2.7) sind leicht nachvollziehbar: da die berechneten Ellipsen jeweils die Bahnpunkte des unverdichteten Orbits als

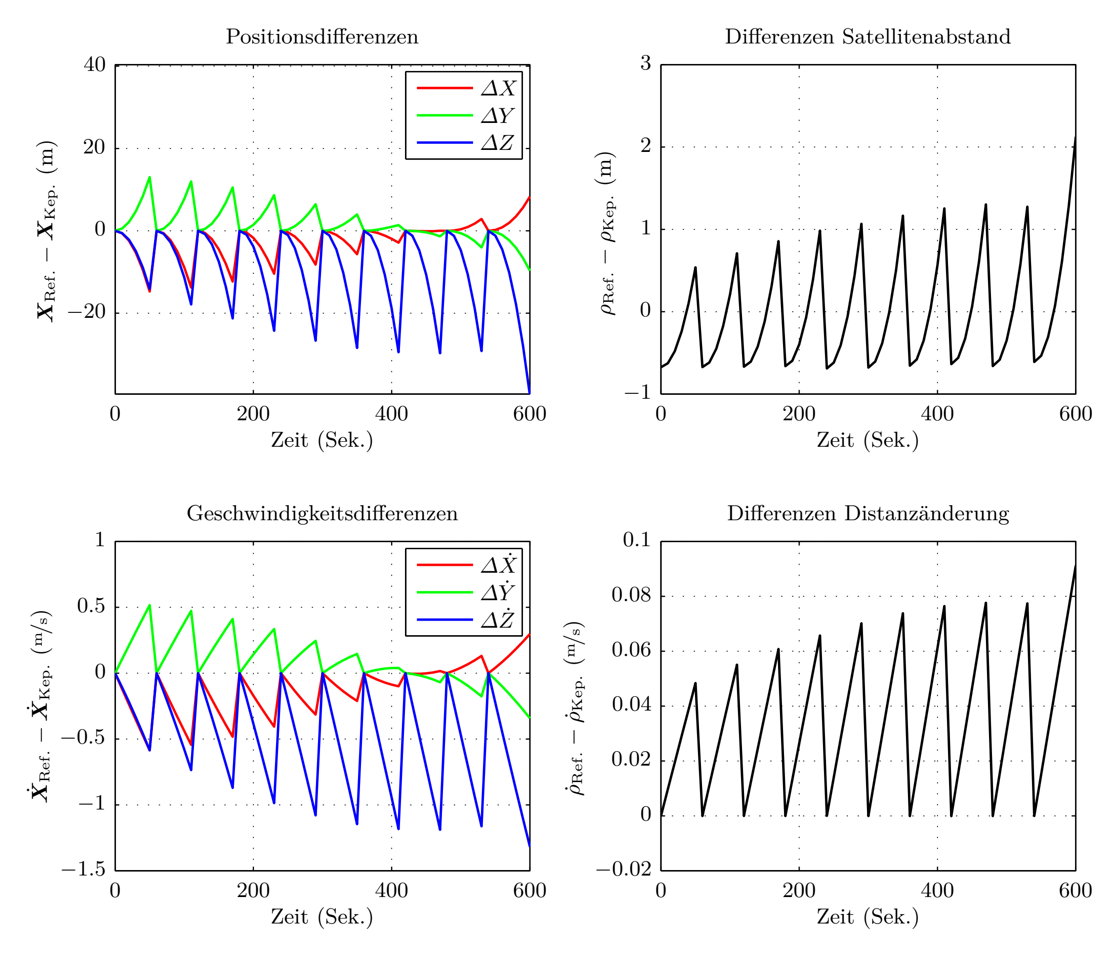


Abb. 2.7: Referenz vs. Verdichtung mit Keplerorbits

Startpunkte haben, sind dort die Abweichungen zur Referenz gerade so groß wie das aufmodulierte Rauschen. Kurz vor dem nächsten Bahnpunkt erreichen die Differenzen maximale Werte. Die Positions- und Geschwindigkeitsfehler werden derart groß, weil die Keplerbewegung (2.27) nur grob die wirkliche Bewegung des Satelliten modelliert. Denn beim ungestörten Zweikörperproblem wird unterstellt, dass ausschließlich die Zentralkraft der als Punktmasse aufgefassten Erde wirke und die Masse des Satelliten vernachlässigbar klein gegenüber der Masse des Zentralkörpers sei, d.h. die Satellitenmasse entspreche der Einheitsmasse $m = 1$.

2.3.4 Verdichtung mittels Schießverfahren

Tatsächlich erfährt der Satellit viele weitere Beschleunigungen, weshalb man die Gleichung der Keplerbewegung (2.27) zu der allgemeineren Form

$$\ddot{\boldsymbol{X}}(t) = \boldsymbol{f}_{\mathrm{K}} = -\frac{GM}{\|\boldsymbol{X}(t)\|^3} \cdot \boldsymbol{X}(t) + \boldsymbol{f}_{\mathrm{S}}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \boldsymbol{p}(t)) \tag{2.28}$$

erweitert (vgl. z.B. Seeber, 1989, S. 82 ff.). Darin ist $\boldsymbol{f}_{\mathrm{K}}$ die sog. Kräftefunktion, die alle Beschleunigungen repräsentiert, welche auf den Satelliten einwirken. Sie kann zerlegt werden in den bekannten Keplerterm und einen Störvektor $\boldsymbol{f}_{\mathrm{S}}$, der die Summe der auftretenden Störbeschleunigungen beinhaltet. Die Störbeschleunigungen werden vor allem verursacht durch

(i) den inhomogenen, nicht kugelförmigen Aufbau des Erdkörpers,

(ii) die Gravitationswirkung von Drittkörpern (Mond, Sonne etc.),

(iii) die Gezeiten der festen Erde und die Ozeangezeiten

(iv) sowie diverse Oberflächenkräfte am Satelliten (z.B. Atmosphärenreibung, Strahlungsdruck der Sonne).

Die Kraftfunktion $\boldsymbol{f}_{\mathrm{K}}$ bzw. der Störanteil $\boldsymbol{f}_{\mathrm{S}}$ ist i.A. nicht-linear von der Zeit, der Position und Geschwindigkeit des Satelliten abhängig sowie von weiteren Parametern, die im Parametervektor $\boldsymbol{p}$ zusammengefasst seien (dazu gehören z.B. die Potentialkoeffizienten des Gravitationsfeldmodells). Die unter Punkt (iv) genannten nicht-gravitativen Störbeschleunigungen können mittels der im Satelliten vorhandenen Beschleunigungssensoren (Akzelerometer) in ihrer Summe gemessen werden und als Messwerte in die Kräftefunktion eingehen. Denn die Oberflächenkräfte lassen sich bislang nur unzureichend bzw. erschwert durch Modelle beschreiben (vgl. Reubelt, 2009, S. 14).

In dieser Arbeit beschränkt man sich allerdings darauf, nur die Störbeschleunigung durch das anisotrope Gravitationsfeld nach Punkt (i) in die Kräftefunktion miteinzubeziehen – aufgrund folgender Überlegungen:

- Zufolge Grafik 3.1 in Montenbruck und Gill (2005, S. 55), welche die Größenordnungen der Störbeschleunigungen illustriert, sind die betragsmäßig größten Beschleunigungen auf die GRACE-Satelliten tatsächlich dem Erdgravitationsfeld (bis etwa Grad und Ordnung 18) zuzurechnen. GRACE gehört mit ca. 460 km, ebenso wie die Iridium-Satelliten (Bahnhöhe ca. 780 km), zur Kategorie der LEO-Satelliten. Nach ebendieser Grafik sind die Atmosphärenreibung, die direkte Gezeiteneinwirkung von Sonne und Mond sowie die indirekten Gezeiteneffekte (Ozeangezeiten und Gezeiten der festen Erde) die nächst größeren Störeinflüsse auf einen solchen Orbit mit geringer Bahnhöhe. Folglich müssten zusätzlich die entsprechenden Störbeschleunigungen durch Modelle in der Kräftefunktion berücksichtigt werden, um die Abweichungen zwischen integrierten und gemessenen Bahnvektoren zu minimieren, d.h. die Genauigkeit der Prädiktion zu verbessern. Wenn keine Akzelerometermessungen zur Verfügung stehen, müsste man dabei auch die Atmosphärenreibung durch ein Modell beschreiben. Dies alles umzusetzen, würde jedoch den für diese Arbeit veranschlagten Arbeitsaufwand übersteigen.
- Wegen der relativ kurzen Integrationszeit von 60 Sekunden bei realen Daten (Dauer zwischen zwei aufeinanderfolgenden GRACE-Navigationslösungen) bleiben die durch die Vernachlässigung der kleineren Störbeschleunigungen eingeführten Fehler (d.h. die Abweichungen zur wahren Trajektorie) ohnehin gering im Vergleich zu einer Integration über längere Zeitintervalle von etwa mehreren Stunden.

Im Falle simulierter Daten sind auch längere Integrationszeiten unproblematisch, sofern man hierbei zur Integration der Referenzorbits und zur Orbitverdichtung (mittels Integrationsverfahren) ein identisches Kräftemodell verwendet.

Es wird nun eine weitere Methode zur Orbitverdichtung untersucht, welche das sogenannte Einfach-Schießverfahren nutzt. Dabei fasst man für den gewählten Satellitenbahnabschnitt von 10 Minuten die sich aus Gl. (2.28) ergebende vektorielle DGL

$$\ddot{\boldsymbol{X}}(t) = \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}(t)\, \nabla_{\mathrm{E}} V(\boldsymbol{x}(t), \boldsymbol{p}) \tag{2.29}$$

als nicht-lineares Randwertproblem (RWP) auf mit den Randbedingungen

$$\boldsymbol{X}(t_0) = \boldsymbol{X}_{\mathrm{nav}}(t_0), \quad \boldsymbol{X}(t_e) = \boldsymbol{X}(t_0 + 10\,\mathrm{Min.}) = \boldsymbol{X}_{\mathrm{nav}}(t_e). \tag{2.30}$$

In dieser Voruntersuchung sollen also die Anfangs- und Endposition des 10-Minuten-Bogens mit den entsprechenden Positionen der simulierten Navigationslösung[1] übereinstimmen. Die zur Bestimmung des verdichteten Orbits erforderliche numerische Integration der Bewegungsgleichung (2.29) im raumfesten System wird später in Abschnitt 3.2.5.1 ausführlich behandelt. Im Folgenden werden zunächst die Grundlagen des Schießverfahrens geschildert.

1 Ortsvektoren des verrauschten, unverdichteten Orbits; hier mit „nav“ gekennzeichnet

Der Zweck des Schießverfahrens ist grundsätzlich, ein Randwertproblem in ein (näherungsweise) äquivalentes Anfangswertproblem (AWP) zu überführen[1]. Somit würde eine numerische Integration des gleichwertigen AWPs genügen, um den verdichteten Orbit unter Einhaltung der Randbedingungen zu erhalten. Dazu ist erforderlich, die zum AWP noch fehlende zweite Anfangsbedingung[2] so zu finden, dass die Lösung des AWPs auch die Randbedingung am rechten Rand (hier $\boldsymbol{X}(t_e) = \boldsymbol{X}_{\text{nav}}(t_e)$) erfüllt.

Zu Beginn der Orbitverdichtung mithilfe des Schießverfahrens ist die DGL (2.29) mit den Anfangsbedingungen $\boldsymbol{X}_{\text{nav}}(t_0)$, $\dot{\boldsymbol{X}}_{\text{nav}}(t_0) =: \dot{\boldsymbol{X}}_{\text{corr}_0}(t_0)$ numerisch zu integrieren, was auf die Lösungen $\boldsymbol{X}_{\text{int}_0}(t_i)$, $\dot{\boldsymbol{X}}_{\text{int}_0}(t_i)$ am Bogenendpunkt führt. Diese Lösung erfüllt i.A. nicht die Randbedingung $\boldsymbol{X}_{\text{nav}}(t_e)$, sondern weist zu dieser Soll-Position die Differenz $\boldsymbol{f}_0 = \boldsymbol{X}_{\text{nav}}(t_e) - \boldsymbol{X}_{\text{int}_0}(t_e)$ auf.

Wird die Anfangssteigung $\dot{\boldsymbol{X}}_{\text{nav}}(t_0)$ leicht variiert, kann man analog eine neue Differenz $\boldsymbol{f}_1$ am rechten Rand berechnen, d.h. es gilt offenbar $\boldsymbol{f} = \boldsymbol{f}(\dot{\boldsymbol{X}}(t_0))$. Bei linearen Randwertproblemen ist diese Abhängigkeit zwischen der Differenz und der Anfangssteigung linear, sodass bereits eine weitere Lösung mit einer variierten zweiten Anfangsbedingung ausreicht, um mithilfe der Linearkombination der beiden berechneten Lösungen einen eindeutigen Anfangsvektor zu berechnen, für welchen die Differenz $f = 0$ folgt. Bei nicht-linearen Randwertproblemen muss hingegen die Nullstelle der implizit definierten Funktion f iterativ gesucht werden. Da die DGL (2.29) aus drei Gleichungen zweiter Ordnung (bzw. sechs Gleichungen erster Ordnung) besteht, liefert jede Lösung des AWPs nicht nur eine Differenz am Bogenendpunkt, sondern drei (eine Differenz pro Positionskomponente):

$$\boldsymbol{f}_j(\dot{\boldsymbol{X}}_{\text{corr}_j}(t_0)) = \left(\boldsymbol{X}_{\text{nav}}(t_e) - \boldsymbol{X}_{\text{int}_j}(t_e)\right)_{3\times 1} = \begin{pmatrix} X_{\text{nav}}(t_e) - X_{\text{int}_j}(t_e) \\ Y_{\text{nav}}(t_e) - Y_{\text{int}_j}(t_e) \\ Z_{\text{nav}}(t_e) - Z_{\text{int}_j}(t_e) \end{pmatrix}. \tag{2.31}$$

Darin bezeichne $j = 0, \ldots, n \in \mathbb{N}$ den Iterationsschritt. Benötigt wird ein iterativer Algorithmus, der die Anfangsbedingungen $\dot{\boldsymbol{X}}_{\text{corr}}(t_0)$ so variiert, dass die Differenzen am Bogenendpunkt gegen Null gehen:

$$\lim_{j\to\infty} \boldsymbol{f}_j(\dot{\boldsymbol{X}}_{\text{corr}_j}(t_0)) = \boldsymbol{0} \tag{2.32}$$

Sobald die Nullstelle hinreichend genau feststeht, wird der Algorithmus bereits nach endlich vielen Schritten n beendet.

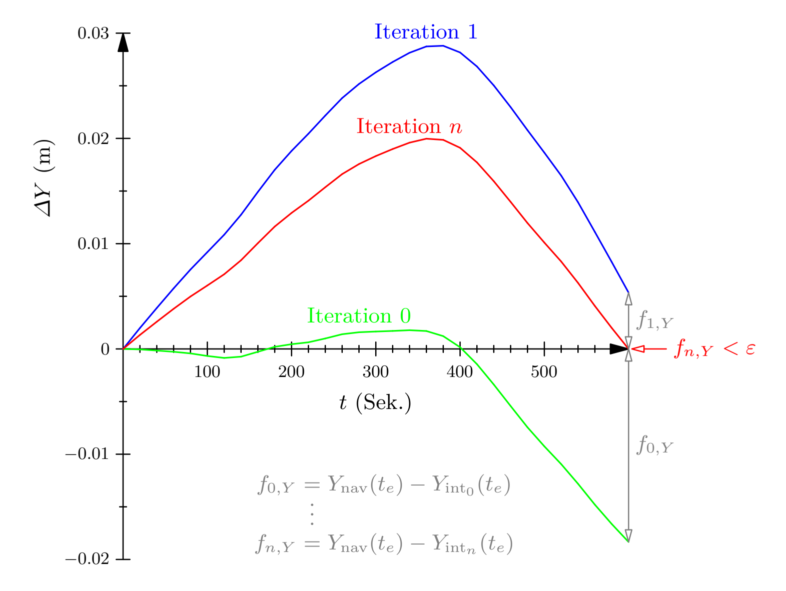


Abb. 2.8: Iteratives Lösungsprinzip beim Schießverfahren für nicht-lineare Randwertprobleme

Abbildung 2.8 zeigt den prinzipiellen Ablauf der iterativen Lösung bezüglich der Y-Komponente. Ausgehend von der Abweichung $\boldsymbol{f}$ am rechten Rand wird jeweils ein korrigierter Anfangsvektor $\dot{\boldsymbol{X}}_{\text{corr}_j}(t_0)$ bestimmt, der – nach einer erneuten Orbitintegration – eine (meist) kleinere Differenz am rechten Rand liefert. Dies wird so lange wiederholt, bis ein bestimmtes Abbruchkriterium erfüllt ist, z.B. könnte die Iteration abgebrochen werden, falls die Differenzen $\boldsymbol{f}$ einen vorgegebenen Schwellwert ε unterschreiten.

1 Zur Theorie des Schießverfahrens siehe z.B. Hermann (2004, S. 233 ff.) und Reinhardt (2008, S. 111 ff.).
2 Da hier als Randbedingungen die gegebenen Positionen dienen, ist die gesuchte „zweite“ Anfangsbedingung der Geschwindigkeitsvektor $\dot{\boldsymbol{X}}(t_0)$, d.h. sozusagen die Steigung bzw. der „Schusswinkel“ im Anfangspunkt.

Eine derartige Lösung nicht-linearer Gleichungssysteme mit der Zielfunktion (2.32) ist in MATLAB mithilfe der Funktion `fsolve` möglich, die Bestandteil der Optimization Toolbox™ ist[1]. Für die eigene Toolbox wurde die Funktion `shootorb` (s.a. Anhang A.2, S. 142) programmiert, um die Orbitverdichtung mittels des Schießverfahrens durchzuführen, wobei ebendiese `fsolve`-Routine zum Einsatz kommt. Weitere Details zum Aufruf und der Verwendung der Funktion `fsolve` findet man im Benutzerhandbuch der Optimization Toolbox™ ab Seite 11-122 (O.T.-HANDBUCH, 2010). Folgende Lösungsalgorithmen sind in `fsolve` implementiert:

- „Trust-region-dogleg"- und „trust-region-reflective"-Methode, siehe beispielsweise COLEMAN und LI (1992) sowie COLEMAN und LI (1996).
- Levenberg-Marquardt-Algorithmus. Dieser wird z.B. ausführlich in MORÉ (1978) behandelt.
- Klassisches Gauss-Newton-Verfahren. Dessen Anwendung beim Schießverfahren ist etwa in REINHARDT (2008, S. 117 ff.) aufgezeigt.

Laut dem Handbuch sind für den Abbruch der Algorithmen folgende Parameter maßgeblich, die mittels `optimset` angepasst und an den Löser (hier die `fsolve`-Funktion) übergeben werden können[2]:

`MaxIter` maximale Anzahl der Iterationen.

`TolFun` untere Grenze für die Änderung der Zielfunktion zwischen den Iterationen. Falls die Norm der Differenz zwischen dem aktuellen Wert der Zielfunktion und dem vorherigen Wert der Zielfunktion kleiner ist als `TolFun`, wird der Algorithmus beendet. Der Abbruch erfolgt also, falls sich sozusagen der Wert der Zielfunktion nicht mehr wesentlich ändert.

`TolX` untere Grenze für die Schrittgrößenänderung zwischen den Iterationen. Hier bedeutet dies, dass die Iteration abgebrochen wird, wenn die Norm der Differenz zwischen der vom Löser aktuell geschätzten Korrektur und der zuletzt geschätzten Korrektur kleiner ist als `TolX`.

Nach einigen Durchläufen (hier nach 4 Iterationen) findet `fsolve` die endgültige Korrektur für die Anfangsbedingungen. Am Ende muss in `shootorb` das AWP mit den korrigierten Anfangsbedingungen nochmals numerisch integriert werden, um den verdichteten Orbit zu erhalten.

Die resultierenden Abweichungen der verdichteten Satellitenbahn zur Referenz in Abbildung 2.9 zeigen einen auffallend glatten Kurvenverlauf. Dies hängt zum einen damit zusammen, dass durch die gewählte lange ‚Schussweite' von 10 Minuten alle dazwischen liegenden Positionen $\boldsymbol{X}_{\text{nav}}$ im Gegensatz zu den Interpolationsverfahren nicht als Stützpunkte genutzt werden, sondern völlig außer Acht bleiben. Zum anderen wurde der Referenzorbit basierend auf demselben Kräftemodell erzeugt, das auch bei der Verdichtung verwendet wurde. Im Falle realer Satellitenbahnen ist aufgrund der Überlagerung vieler Einzelstörungen mit einem unstetigeren Verlauf zu rechnen. Bezüglich der Relativgeschwindigkeit $\dot{\rho}$ liegt die Abweichung zur Referenz hier in der Größenordnung um 10^{-5} m/s (s. Abb. 2.9), was eine deutliche Verbesserung gegenüber den bisherigen Orbitverdichtungsverfahren, aber nur eine geringfügige Verbesserung gegenüber der simulierten Navigationslösung bedeutet. Dies ist anhand Tabelle 2.2 ersichtlich, vergleicht man die Standardabweichungen des Schießverfahrens mit den entsprechenden Werten der simulierten Navigationslösung bzw. der anderen Verfahren.

Insgesamt scheitern alle untersuchten einfachen Verfahren an den hohen Anforderungen, denn die Abweichungen zu den Referenzgeschwindigkeiten und zur K-Band-Größe $\dot{\rho}$ sind noch zu groß. Wünschenswert wäre eine Steigerung der Genauigkeit hinsichtlich $\Delta\dot{\rho}$ auf eine Größenordnung $< 10^{-6}$ m/s.

1 http://www.mathworks.com/products/optimization/
2 Vgl. O.T.-HANDBUCH (2010, S. 2-43)

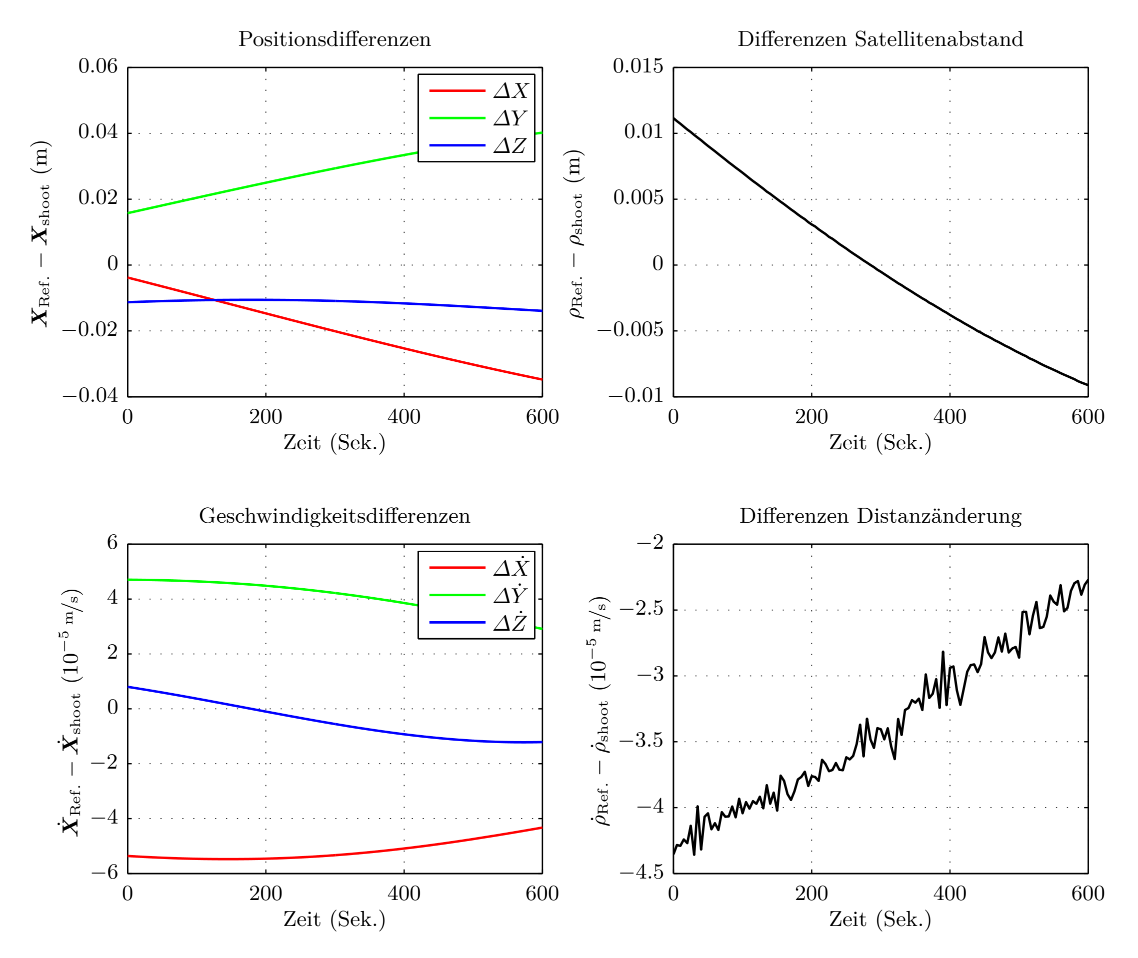


Abb. 2.9: Referenz vs. Verdichtung nach dem Schießverfahren

Tab. 2.2: Fehlertheoretische Gegenüberstellung der einfachen Orbitverdichtungsverfahren

		Navigations-lösung	Lineare Verdichtung	Kubische Splines	Keplerorbits	Schießverfahren
$\max\{\|\Delta X_i\|\}$	(m)	0.035	1192.658	1.071	14.743	0.035
$\max\{\|\Delta Y_i\|\}$	(m)	0.040	1055.279	0.937	13.043	0.040
$\max\{\|\Delta Z_i\|\}$	(m)	0.034	3797.668	3.703	40.020	0.014
σ_X	(m)	0.016	476.099	0.263	4.751	0.022
σ_Y	(m)	0.018	419.015	0.239	4.231	0.030
σ_Z	(m)	0.016	2702.736	1.208	13.969	0.012
$\max\{\|\Delta \dot{X}_i\|\}$	(m/s)	$3.68{\cdot}10^{-5}$	3.043	$2.96{\cdot}10^{-3}$	0.585	$5.48{\cdot}10^{-5}$
$\max\{\|\Delta \dot{Y}_i\|\}$	(m/s)	$4.01{\cdot}10^{-5}$	2.905	$2.81{\cdot}10^{-3}$	0.516	$4.70{\cdot}10^{-5}$
$\max\{\|\Delta \dot{Z}_i\|\}$	(m/s)	$3.36{\cdot}10^{-5}$	1.751	$1.48{\cdot}10^{-3}$	1.319	$1.22{\cdot}10^{-5}$
$\sigma_{\dot{X}}$	(m/s)	$1.72{\cdot}10^{-5}$	2.162	$9.53{\cdot}10^{-4}$	0.221	$5.18{\cdot}10^{-5}$
$\sigma_{\dot{Y}}$	(m/s)	$2.07{\cdot}10^{-5}$	2.073	$9.22{\cdot}10^{-4}$	0.195	$4.12{\cdot}10^{-5}$
$\sigma_{\dot{Z}}$	(m/s)	$1.90{\cdot}10^{-5}$	0.695	$3.70{\cdot}10^{-4}$	0.641	$7.84{\cdot}10^{-6}$
$\max\{\|\Delta \rho_i\|\}$	(m)	0.053	124.208	0.127	2.125	0.011
$\max\{\|\Delta \dot{\rho}_i\|\}$	(m/s)	$8.32{\cdot}10^{-5}$	$2.18{\cdot}10^{-3}$	$8.41{\cdot}10^{-5}$	$9.12{\cdot}10^{-2}$	$4.36{\cdot}10^{-5}$
σ_ρ	(m)	0.024	79.350	0.042	0.680	0.006
$\sigma_{\dot{\rho}}$	(m/s)	$4.59{\cdot}10^{-5}$	$1.33{\cdot}10^{-3}$	$4.41{\cdot}10^{-5}$	$4.33{\cdot}10^{-2}$	$3.44{\cdot}10^{-5}$

Kapitel 3

3 Theoretische Ausarbeitung

Um das Bewegungsverhalten eines Vorgangs in der realen Welt zu beschreiben, kann oftmals ein *dynamisches System* angenommen werden. Generell wird dabei eine Eingangsgröße (skalare oder vektorielle Zeitfunktion) in eine Ausgangsgröße (den modellierten Systemzustand) transformiert, wobei jedoch ein einzelner Momentanwert der Eingangsgröße nicht ausreicht, um die aktuelle Ausgangsgröße zu bestimmen, sondern auch frühere Eingangsinformationen miteinzubeziehen sind (vgl. Schneeweiss, 1974, S. 1). Dynamische Systeme (auch: Systeme mit konzentrierten Parametern, engl.: „lumped systems“) werden durch Differenzengleichungen oder gewöhnliche Differentialgleichungen erster Ordnung beschrieben. Die Anzahl der Gleichungen entspricht hierbei der minimalen Zahl von Anfangsbedingungen, die das künftige Systemverhalten eindeutig festlegen (Schrick, 1977, S. 311). Ein künstlicher Erdsatellit kann demnach als ein dynamisches System aufgefasst werden, da sich sein Bewegungsverhalten mithilfe einer DGL (siehe Gl. (2.28)) modellieren lässt.

Derartige Modelle stellen stets mathematisch, physikalisch und technisch sinnvolle Abstraktionen der Wirklichkeit dar und enthalten zwangsläufig Vereinfachungen aufgrund von Näherungen, unbekannten Parametern/Eingangsgrößen oder nicht modellierbaren Effekten, sodass das gemessene reale Systemverhalten vom modellierten Systemverhalten in nicht vorhersehbarer Weise abweichen kann (Loffeld, 1990, S. 3).

Die erforderlichen Messungen am realen System sind selbst ebenfalls fehlerbehaftet, da bekanntlich jede Beobachtung mit gewissen Ungenauigkeiten verbunden ist. Verschiedene Algorithmen der Schätztheorie – und dazu gehört insbesondere das Kalman-Filter – haben die Aufgabe, trotz der vorhandenen Modellierungs- und Messfehler eine bestmögliche Schätzung des Systemzustandes und ggf. weiterer interessierender Größen zu ermöglichen (siehe Abb. 3.1). Die erwähnten Störungen und Fehler werden also sozusagen ‚herausgefiltert‘ bzw. unterdrückt.

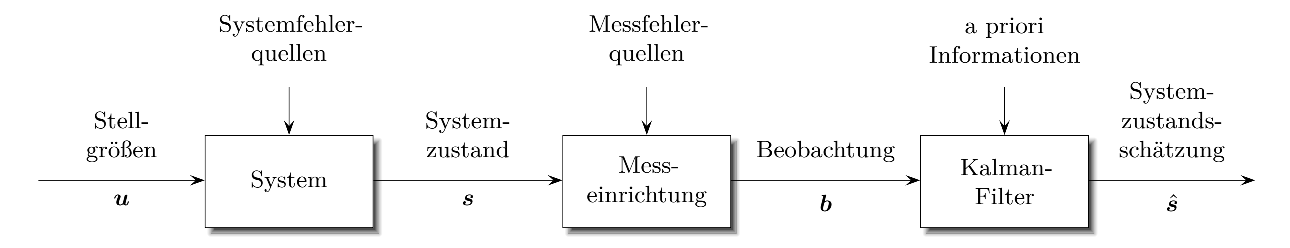


Abb. 3.1: Anwendung des Kalman-Filters zur Zustandsschätzung dynamischer Systeme (nach Gelb, 1996, S. 3; Grewal und Andrews, 2008, S. 114; Loffeld, 1990, S. 4; Maybeck, 1979, S. 5; Schrick, 1977, S. 24)

Ursprünglich hat Rudolph Emil Kalman das heute unter seinem Namen bekannte Filter für die optimale Zustandsschätzung eines *linearen* dynamischen Systems (lineare und zeitdiskrete System- und Messgleichung) entworfen (Kalman, 1960). Ein Jahr später veröffentlichte er gemeinsam mit Richard Snowden Bucy die Theorie des Filters für zeitkontinuierliche, lineare Schätzprobleme, welches daher häufig als Kalman-Bucy-Filter bezeichnet wird (Kalman und Bucy, 1961). Später kamen „erweiterte“ Filtervarianten für Systeme mit nicht-linearen Gleichungen hinzu (z.B. das

„*Extended Kalman Filter*“, EKF). Laut McGee und Schmidt (1985, S. 5) hatte einst Stanley F. Schmidt bei einer Besprechung mit R. E. Kalman die Idee zu dem Linearisierungsansatz, auf dem das EKF basiert.

Durch den wiederholten Vorgang der Vorhersage des Systemzustandes und der Korrektur desselben, sobald Beobachtungen vorliegen, ist das Kalman-Filter sowohl bei Echtzeit-Auswertungen als auch im Postprocessing anwendbar. Vorteilhaft ist außerdem, dass unterschiedliche Messungen bzw. Sensordaten für diese Korrektur genutzt werden können. Das Konzept der Kalman-Filterung hat sich somit bereits in vielen Anwendungsbereichen – z.B. bei Navigationsaufgaben, in der Radar- und Regelungstechnik und auch in der Geodäsie – als äußerst leistungsfähig erwiesen.

Im Folgenden wird zunächst die prinzipielle Arbeitsweise der Kalman-Filterung anhand des linearen, zeitdiskreten Falles aufgezeigt. Für die nicht-lineare Problemstellung der Orbitverdichtung nutzt man die danach beschriebene Technik der erweiterten Kalman-Filterung (EKF).

Vorweg sei angemerkt, dass hier nur einige grundlegende und praktische Aspekte der Kalman-Filterung beschrieben werden können. Für weiterführende Betrachtungen der Thematik, wie insbesondere die mathematischen Herleitungen, sei z.B. verwiesen auf Brown und Hwang (1997); Chui und Chen (2009); Gelb (1996); Grewal und Andrews (2008); Loffeld (1990); Maybeck (1979, 1982a,b); Schneeweiss (1974); Schrick (1977); Simon (2006). In der genannten Literatur zur Kalman-Filterung wird der Vektor des Systemzustands gewöhnlich mit $\boldsymbol{x}$ und der Messwert- bzw. Beobachtungsvektor mit $\boldsymbol{y}$ oder $\boldsymbol{z}$ benannt. Um Verwechslungen mit dem bereits definierten erdfesten Positionsvektor oder dessen Komponenten zu vermeiden, bezeichne hier $\boldsymbol{s}$ den Zustandsvektor und $\boldsymbol{b}$ den Vektor der Messungen. Ansonsten entsprechen die gewählten Bezeichnungen und Symbole für die Kalman-Filterung der üblicherweise verwendeten Nomenklatur.

3.1 Grundlagen der linearen, zeitdiskreten Kalman-Filterung

3.1.1 Systemmodell

Das betrachtete dynamische System genüge der vektoriellen, stochastischen Differenzengleichung

$$\boldsymbol{s}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{s}_k + \boldsymbol{B}_k \boldsymbol{u}_k + \boldsymbol{G}_k \boldsymbol{w}_k, \tag{3.1}$$

die den n-dimensionalen Zustandsvektor $\boldsymbol{s}_k$ vom Zeitpunkt t_k auf den darauffolgenden Zeitpunkt t_{k+1} überträgt ($k \in \mathbb{N}_0$). Der Vektor des Systemzustands $\boldsymbol{s}$ enthält im Allgemeinen die n interessierenden geometrischen oder physikalischen Größen. Ferner bezeichnet

- $\boldsymbol{\Phi}_k$ die quadratische $(n \times n)$-Transitionsmatrix (auch Übergangs-/Systemmatrix), welche individuell für jede Problemstellung gemäß Gl. (3.12) zu formulieren ist,
- $\boldsymbol{u}_k$ den p-dimensionalen Vektor der Stellgrößen (auch Steuereingangsgrößen), welche das System zusätzlich deterministisch beeinflussen[1],
- $\boldsymbol{B}_k$ die $(n \times p)$-Stellgrößenmatrix (auch Steuereingangsmatrix), welche die lineare Abhängigkeit der Zustandsgrößen von den Stellgrößen zum Zeitpunkt t_k beschreibt.

Während die beiden ersten Summanden in (3.1) den deterministischen und damit größten Bestandteil der Systemgleichung darstellen, modelliert der letzte Term den stochastischen Einfluss, der sich zusammensetzt aus:

1 Bei Navigationsaufgaben kann dies z.B. der Lenkwinkel eines Fahrzeugs sein, im Falle eines Satelliten etwa Zustandsänderungen durch Orbitmanöver (vgl. Schäfer, 2000, S. 62)

$\boldsymbol{w}_k$ r-dimensionaler Vektor der Störgrößen (auch: Vektor des Prozess-/Systemrauschens), welcher zum einen eventuelle Systemstörungen und zum anderen nicht funktional modellierte Eingangsgrößen (Unsicherheit des deterministischen Modells) berücksichtigen soll.

$\boldsymbol{G}_k$ $(n \times r)$-Störgrößenmatrix (auch Einflussmatrix), gibt den linearen Zusammanhang zwischen den Störgrößen und den Zustandsgrößen an.

Für die Störgröße $\boldsymbol{w}_k$ wird hier mittelwertfreies, normalverteiltes weißes Rauschen angenommen. Die Zufallsvariablen dieses Rauschprozesses weisen bekanntlich zu jedem Zeitpunkt eine identische Verteilung auf (konstante spektrale Leistungsdichte) und sind daher zeitlich nicht miteinander korreliert. Durch Bildung des Erwartungswertes folgt als zweites zentrales Moment die $(r \times r)$-Kovarianzmatrix des Systemrauschens, welche positiv semidefinit sei und mit $\boldsymbol{Q}_k$ bezeichnet wird:

$$\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k) \tag{3.2}$$

$$\mathrm{E}\{\boldsymbol{w}_j \boldsymbol{w}_k^\top\} = \begin{cases} \boldsymbol{Q}_k & j = k \\ \boldsymbol{0} & j \neq k. \end{cases} \tag{3.3}$$

3.1.2 Beobachtungsmodell

Der Systemzustand $\boldsymbol{s}_k$ ist nicht direkt physikalisch messbar, sondern sei nur eine ‚verrauschte' Linearkombination der Komponenten von $\boldsymbol{s}_k$. Die zugrundegelegte Beobachtungsgleichung

$$\boldsymbol{b}_k = \boldsymbol{H}_k \boldsymbol{s}_k + \boldsymbol{v}_k \tag{3.4}$$

gilt für alle $k \in \mathbb{N}$, die erste Beobachtung erfolgt also zum Zeitpunkt t_1. Hierin bezeichnet

$\boldsymbol{b}_k$ den m-dimensionalen Mess-/Beobachtungsvektor zum Zeitpunkt t_k,

$\boldsymbol{H}_k$ die $(m \times n)$-Messmatrix, welche den Systemzustand linear in den Messraum abbildet,

$\boldsymbol{s}_k$ den unbekannten n-dimensionalen Zustandsvektor,

$\boldsymbol{v}_k$ den $(m \times 1)$-Vektor der Messfehler (bzw. des Messrauschens).

Das Messrauschen $\boldsymbol{v}_k$ wird ebenfalls als erwartungswertfreie, gaußverteilte weiße Zufallsvariable modelliert, deren symmetrische und positiv definite $(m \times m)$-Kovarianzmatrix mit $\boldsymbol{R}_k$ benannt sei:

$$\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}_k) \tag{3.5}$$

$$\mathrm{E}\{\boldsymbol{v}_j \boldsymbol{v}_k^\top\} = \begin{cases} \boldsymbol{R}_k & j = k \\ \boldsymbol{0} & j \neq k. \end{cases} \tag{3.6}$$

Zudem soll $\boldsymbol{v}_k$ stochastisch unabhängig vom Systemrauschen $\boldsymbol{w}_k$ sein, d.h. für die Kreuzkorrelation der beiden Rauschsequenzen fordert man

$$\mathrm{E}\{\boldsymbol{v}_j \boldsymbol{w}_k^\top\} = \boldsymbol{0} \quad \forall\, j, k \in \mathbb{N}_0. \tag{3.7}$$

Bei der Zustandsraummodellierung wird außerdem die Annahme getroffen, dass der Ausgangs- oder Bezugszustand zum Zeitpunkt t_0 und die zugehörige Schätzfehlerkovarianz durch

$$\mathrm{E}\{\boldsymbol{s}_0\} = \hat{\boldsymbol{s}}_0 \tag{3.8}$$

$$\mathrm{E}\{(\boldsymbol{s}_0 - \hat{\boldsymbol{s}}_0)(\boldsymbol{s}_0 - \hat{\boldsymbol{s}}_0)^\top\} = \widehat{\boldsymbol{P}}_0 \tag{3.9}$$

gegeben sind (Anfangsbedingungen für Gleichung (3.1)), wobei $\hat{\boldsymbol{s}}_0$ eine Schätzung des Anfangszustands $\boldsymbol{s}_0$ darstellt und die symmetrische, positiv definite $(n \times n)$-Matrix $\widehat{\boldsymbol{P}}_0$ die Varianzen und Kovarianzen des geschätzten Anfangszustands (Schätzfehler) enthält. Ferner wird vorausgesetzt, dass der Anfangszustand stochastisch unabhängig von *zukünftigem* Mess- und Systemrauschen ist:

$$\mathrm{E}\{\boldsymbol{w}_k \boldsymbol{s}_0^\top\} = \boldsymbol{0} \qquad \text{und} \qquad \mathrm{E}\{\boldsymbol{v}_k \boldsymbol{s}_0^\top\} = \boldsymbol{0} \quad \forall\, k \in \mathbb{N}\backslash\{0\} \tag{3.10}$$

Es lässt sich nachweisen, dass damit auch die gegenseitigen Korrelationen

$$\mathrm{E}\{\boldsymbol{s}_k \boldsymbol{w}_l^\top\} = \mathrm{E}\{\boldsymbol{s}_k \boldsymbol{v}_l^\top\} = \boldsymbol{0} \quad \text{für} \quad k \leq l \tag{3.11}$$

verschwinden (s. SCHNEEWEISS, 1974, S. 223), d.h. der Systemzustand ist nicht beeinflusst durch späteres System- und Messrauschen.

Die hier getroffenen Annahmen sind zur Herleitung der nachfolgenden Filtergleichungen erforderlich. Einige Annahmen können gelockert werden, dann ergeben sich allerdings andere Filtergleichungen. Beispielsweise ließe sich (entgegen der Annahme in Gl. (3.7)) eine Korrelation zwischen System- und Messrauschen einführen (vgl. SIMON, 2006, Abschn. 7.1) oder man könnte anstelle des weißen von einem farbigen System- und Messrauschen ausgehen (SIMON, 2006, Abschn. 7.2). Derartige Anpassungen könnten auch beim später verwendeten EKF sinnvoll sein und bessere Ergebnisse liefern; in dieser Arbeit beschränkt man sich jedoch darauf, die einfacheren Standardgleichungen (des EKFs) anzuwenden.

3.1.3 Filtergleichungen und prinzipieller Berechnungsablauf

Das Deutsche Institut für Normung definiert das Kalman-Filter in einem Norm-Entwurf[1] als rekursives Filter, das durch die Verbindung von System- und Messgleichung die bestmögliche Zustandsschätzung $\hat{\boldsymbol{s}}$ eines Systems liefert.

Für obige lineare, zeitdiskrete Systembeschreibung ist diese Schätzung des Systemzustandes zu einem beliebigen aktuellen Zeitpunkt[2] *optimal* im Sinne einer minimalen Schätzfehlervarianz (s. unten, Gl. (3.19)). Um diese Aufgabe zu erfüllen, sind bei einer herkömmlichen Ausgleichung nach kleinsten Quadraten alle bis zu diesem Zeitpunkt vorliegenden Informationen, d.h.

- der Ausgangszustand und dessen Kovarianz,
- das a priori Systemmodell,
- sämtliche Beobachtungen
- sowie alle statistische Größen (Messunsicherheiten, Systemstörungen),

mit in die Ausgleichung einzubeziehen, weshalb bei langen Beobachtungsreihen ein enormer Rechenaufwand entsteht. Praktischerweise verfügt das Kalman-Filter jedoch über eine rekursive Prädiktor-Korrektor-Struktur, sodass im Grunde jeweils nur der vorhergehende Systemzustand und die neueste Messung für die Bestimmung des aktuellen Systemzustands zu verwenden sind. Denn wie im Folgenden deutlich wird, ergibt sich jede neue Systemzustandsschätzung aus einer Linearkombination des letzten Schätzwertes und eines Korrekturterms, der von der aktuellen Beobachtung und deren Fehler(ko)varianzen abhängt. Somit sind bei der aktuellen Zustandsschätzung implizit alle bis zu diesem Zeitpunkt eingegangenen Informationen mitberücksichtigt.

Die Funktionsweise der Kalman-Filterung lässt sich in zwei Grundschritte einteilen: der *Prädiktion* des Systemzustandes (engl.: „time update“) und der *Korrektur* des prädizierten Zustandes *mithilfe von Messungen* (engl.: „measurement update“).

1 DIN 18709-5 (Mai 2006): *Begriffe, Kurzzeichen und Formelzeichen in der Geodäsie – Teil 5: Auswertung kontinuierlicher Messreihen*

2 Der Zeitpunkt darf auch zwischen zwei Messungen liegen.

Generell ist bei der Prädiktion aus den gegebenen Eingangsdaten zum Zeitpunkt t_k (Zustandsvektor und Kovarianzmatrix) der weitere Verlauf des Vorganges bis zum Zeitpunkt t_{k+1} der nächsten Messung vorherzusagen. In Analogie zur Systemgleichung (3.1) bestimmt man hierzu den propagierten (a priori) Systemzustand $\bar{\boldsymbol{s}}_{k+1}$ nach Gleichung (3.13), worin die Transitionsmatrix die zeitliche Änderung des prädizierten Systemzustandes in Bezug zum Ausgangszustand aufgrund der deterministisch modellierten Systemdynamik beschreibt:

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \left. \frac{\partial \bar{\boldsymbol{s}}(t)}{\partial \hat{\boldsymbol{s}}_k} \right|_{t=t_{k+1}} . \tag{3.12}$$

Bei linearen Bewegungsgleichungen $\bar{\boldsymbol{s}}(t) = \boldsymbol{f}(t, \hat{\boldsymbol{s}}_k)$ kann für diese Matrix in der Regel ein analytischer Ausdruck gefunden werden. Bildet man den Erwartungswert $\mathrm{E}\{\bar{\boldsymbol{s}}_{k+1}\bar{\boldsymbol{s}}_{k+1}^\top\}$, erhält man die Beziehung (3.14) für die zugehörige Fehlerkovarianzmatrix $\bar{\boldsymbol{P}}_{k+1}$. Darin beschreibt der Term $\boldsymbol{\Phi}_k\widehat{\boldsymbol{P}}_k\boldsymbol{\Phi}_k^\top$ die zeitliche Änderung der Kovarianzmatrix $\widehat{\boldsymbol{P}}_k$ gemäß der Systemdynamik. Das Systemrauschen $\boldsymbol{w}_k$ entfällt bei der Zustandsprädiktion, da es erwartungswertfrei ist und nur stochastisch durch eine Vergrößerung der Fehlerkovarianzmatrix um den Term $\boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^\top$ modelliert wird. Häufig ist keine Stellgröße zur Systembeschreibung notwendig ($\boldsymbol{u}_k \equiv 0$) und das Systemrauschen wirkt üblicherweise direkt im Zustandsraum ($\boldsymbol{G}_k = \mathbf{I} \quad \forall\, k \in \mathbb{N}_0$), wodurch sich die Prädiktionsgleichungen noch ein wenig vereinfachen.

Tafel 3.1: Gleichungen der zeitdiskreten Kalman-Filterung

Schritt 1 – Prädiktion

Voraussage des Systemzustands und der Schätzfehlerkovarianzmatrix:

$$\bar{\boldsymbol{s}}_{k+1} = \boldsymbol{\Phi}_k\hat{\boldsymbol{s}}_k + \boldsymbol{B}_k\boldsymbol{u}_k \tag{3.13}$$

$$\bar{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k\widehat{\boldsymbol{P}}_k\boldsymbol{\Phi}_k^\top + \boldsymbol{B}_k\mathrm{E}\{\boldsymbol{u}_k\boldsymbol{u}_k^\top\}\boldsymbol{B}_k^\top + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^\top \tag{3.14}$$

Startwerte: $\hat{\boldsymbol{s}}_0, \widehat{\boldsymbol{P}}_0$

Schritt 2 – Korrektur durch Messungen

Innovation:

$$\boldsymbol{d}_{k+1} = \boldsymbol{b}_{k+1} - \boldsymbol{H}_{k+1}\bar{\boldsymbol{s}}_{k+1} \tag{3.15}$$

Kalman-Verstärkungsmatrix:

$$\boldsymbol{K}_{k+1} = \bar{\boldsymbol{P}}_{k+1}\boldsymbol{H}_{k+1}^\top \left[\boldsymbol{H}_{k+1}\bar{\boldsymbol{P}}_{k+1}\boldsymbol{H}_{k+1}^\top + \boldsymbol{R}_{k+1}\right]^{-1} \tag{3.16}$$

Schätzung des Systemzustands und der Kovarianzmatrix:

$$\hat{\boldsymbol{s}}_{k+1} = \bar{\boldsymbol{s}}_{k+1} + \boldsymbol{K}_{k+1}\boldsymbol{d}_{k+1} = \bar{\boldsymbol{s}}_{k+1} + \boldsymbol{\delta}\hat{\boldsymbol{s}}_{k+1} \tag{3.17}$$

$$\widehat{\boldsymbol{P}}_{k+1} = \bar{\boldsymbol{P}}_{k+1} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1}\bar{\boldsymbol{P}}_{k+1} \tag{3.18}$$

Hat man auf diese Weise den (a priori) Zustandsvektor $\bar{\boldsymbol{s}}_{k+1}$ und dessen Kovarianzmatrix $\bar{\boldsymbol{P}}_{k+1}$ vorausgesagt, wird im nächsten Schritt, dem Korrekturschritt, zunächst die Differenz $\boldsymbol{d}_{k+1}$ zwischen dem aus der tatsächlichen Messung zum Zeitpunkt t_{k+1} eingegangenen Messvektor $\boldsymbol{b}_{k+1}$ und dem aufgrund der Vorhersage ‚erwarteten' Beobachtungsvektor gebildet, siehe Gl. (3.15). Die so erhaltenen Residuen werden auch als „Innovation" bezeichnet und stellen ein Maß für die Verträg-

lichkeit von theoretischem Systemmodell und den am realen System durchgeführten Messungen dar (EICHHORN, 2005, S. 27).

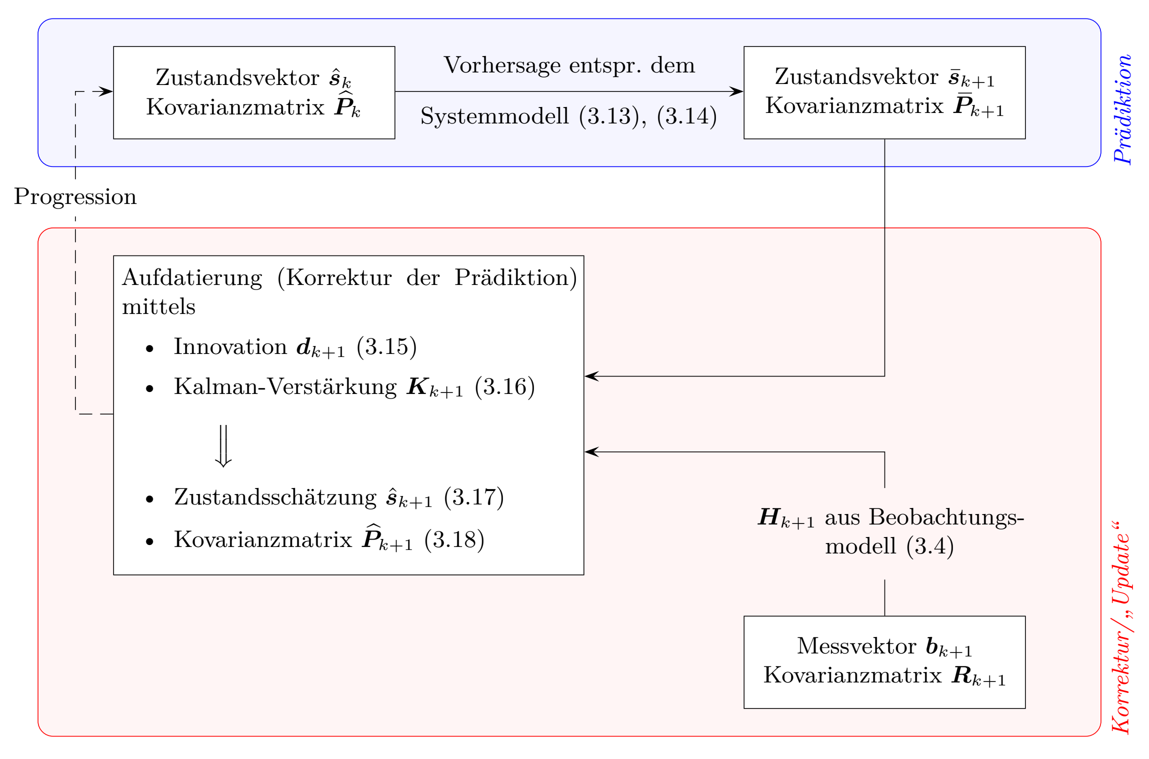


Abb. 3.2: Regelkreis der Kalman-Filterung

Mithilfe dieser Rückmeldung erhält man nun einen verbesserten (a posteriori) Zustandsvektor $\hat{\boldsymbol{s}}_{k+1}$, indem ebendiese Innovation $\boldsymbol{d}_{k+1}$ mit der sog. Kalman-Verstärkungsmatrix $\boldsymbol{K}_{k+1}$ gewichtet und zum prädizierten (a priori) Zustandsvektor $\bar{\boldsymbol{s}}_{k+1}$ addiert wird, siehe Gl. (3.17). Die Kalman-Matrix leistet dabei zweierlei: einerseits nimmt sie eine Transformation des Innovationsvektors vom Beobachtungsraum in den Raum der Zustandsgrößen (Zustandsraum) vor, andererseits legt sie den Einfluss der Innovation auf das Schätzungsergebnis fest. Letztgenannte Gewichtung erfolgt im Grunde entsprechend den Fehlern (Kovarianzmatrizen) des vorhergesagten und des ‚beobachteten‘ Systemzustands: die Korrektur wird stark gewichtet, wenn die Messung exakt und die Prädiktion vergleichsweise unsicher ist, umgekehrt sind die Einträge der Verstärkungsmatrix klein, falls die Prädiktion zuverlässiger ist als die Beobachtung. Im Extremfall $\boldsymbol{K} = \boldsymbol{0}$ würde sich das Filter wie ein Prädiktor verhalten, bei dem die Zustandsschätzwerte nur auf einer Voraussage beruhen, ohne Ausnutzung von Messungen (vgl. SCHRICK, 1977, S. 43). Um eine optimale Gewichtung zu erreichen, fordert man, dass die Summe der resultierenden Schätzfehlervarianzen möglichst klein sein soll. Die entsprechende Zielfunktion hierfür lautet

$$\mathrm{tr}(\widehat{\boldsymbol{P}}_{k+1}) \longrightarrow \min. \tag{3.19}$$

Mit diesem Kriterium lässt sich die Vorschrift (3.16) zur Berechnung der Kalman-Verstärkungsmatrix herleiten (u.a. SCHRICK, 1977, S. 58). Es kann gezeigt werden, dass die so gewählte Verstärkungsmatrix nicht nur die Summe der Schätzfehlervarianzen, sondern darüberhinaus auch jede einzelne Schätzfehlervarianz minimiert. Außerdem gilt für den Erwartungswert des Schätzfehlers $\mathrm{E}\|\boldsymbol{\varepsilon}_k\|^2 = 0$ mit $\boldsymbol{\varepsilon}_k = \boldsymbol{s}_k - \hat{\boldsymbol{s}}_k$. Die Schätzung (3.17) ist also erwartungstreu. Weitere Eigenschaften des diskreten Kalman-Filters sind etwa in WENDEL (2007, S. 139) aufgeführt. Anhand Gleichung (3.18) ist sofort erkennbar, dass infolge des Korrekturschritts ein Genauigkeitsgewinn zu verzeichnen ist, denn die aufdatierte Kovarianzmatrix ist offensichtlich kleiner als die Kofaktormatrix des vorhergesagten Zustandsvektors. Bei der Implementierung des ‚Kovarianz-Updates‘ ist

allerdings die mathematisch äquivalente *Joseph*-Form[1]

$$\widehat{\boldsymbol{P}}_{k+1} = (\mathbf{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1})\overline{\boldsymbol{P}}_{k+1}(\mathbf{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1})^{\top} + \boldsymbol{K}_{k+1}\boldsymbol{R}_{k+1}\boldsymbol{K}_{k+1}^{\top} \tag{3.20}$$

zu bevorzugen, da sie numerisch robuster ist und die Symmetrie sowie die positive Definitheit von $\widehat{\boldsymbol{P}}_{k+1}$ gewährleistet, sofern diese Eigenschaften auch auf $\overline{\boldsymbol{P}}_{k+1}$ zutreffen (Simon, 2006, S. 129).

Falls nach dem Zeitpunkt t_{k+1} noch eine Messung hinzukommt, wird ein erneuter Filterdurchlauf gestartet, indem der korrigierte Zustandsvektor und dessen Kovarianzmatrix als Startwerte für die Prädiktion bis zum nächsten Messzeitpunkt $t_{k+2} = t_{k+1} + \Delta t$ gesetzt werden. Folglich ergibt sich der in Abbildung 3.2 illustrierte Regelkreis der Kalman-Filterung, der sich durch den wiederholenden Vorgang der beiden Filterschritte – der Prädiktion und der Korrektur mithilfe von Messungen – auszeichnet.

3.2 Orbitverdichtung mittels erweiterter Kalman-Filterung

3.2.1 System- und Beobachtungsmodell

Das Ziel ist nun, das Konzept der Kalman-Filterung zur Verdichtung eines Satellitenorbits zu nutzen. Für den Prädiktionsschritt ist ein dynamisches Modell (Systemmodell) erforderlich, welches das zeitliche Verhalten des Zustandsvektors und damit der Satellitenbewegung beschreibt. Entgegen den Voraussetzungen des bisher betrachteten Kalman-Filters ist dies mithilfe von *linearen* Differenzen- oder Differentialgleichungen *nicht* möglich (bzw. nicht hinreichend genau). Denn bereits die einfache Keplerbewegung liegt entsprechend Gl. (2.27) als gewöhnliches Differentialgleichungssystem zweiter Ordnung vor, das bezüglich $\boldsymbol{X}(t)$ nicht linear ist:

$$\ddot{\boldsymbol{X}}(t) = -\frac{GM}{\|\boldsymbol{X}(t)\|^3} \cdot \boldsymbol{X}(t).$$

Unter Einbeziehung der Geschwindigkeiten ergibt sich daraus das äquivalente System

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \\ \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{pmatrix} = \begin{pmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \\ -\dfrac{GM}{(X^2(t)+Y^2(t)+Z^2(t))^{3/2}} \cdot X(t) \\ -\dfrac{GM}{(X^2(t)+Y^2(t)+Z^2(t))^{3/2}} \cdot Y(t) \\ -\dfrac{GM}{(X^2(t)+Y^2(t)+Z^2(t))^{3/2}} \cdot Z(t) \end{pmatrix} \tag{3.21}$$

mit sechs gewöhnlichen DGLen erster Ordnung (s. z.B. Vermeer, 2010, S. 17).

1 Benannt nach Peter D. Joseph (Space Technology Laboratories, Los Angeles), der die Gleichung hergeleitet und untersucht hat, vgl. Bucy und Joseph (1968).

Allgemeiner geschrieben, erhält man als Systemmodell

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \\ \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{pmatrix} = \begin{pmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \\ f_{\mathrm{K},X}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \boldsymbol{p}(t)) \\ f_{\mathrm{K},Y}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \boldsymbol{p}(t)) \\ f_{\mathrm{K},Z}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \boldsymbol{p}(t)) \end{pmatrix} + \boldsymbol{w}(t). \tag{3.22}$$

Darin bezeichnet $\boldsymbol{f}_{\mathrm{K}}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \boldsymbol{p}(t))$ die spezifische, in den abhängigen Variablen nicht-lineare Kräftefunktion, die alle deterministisch modellierten Beschleunigungen enthält, welche auf den Satelliten einwirken. $\boldsymbol{f}_{\mathrm{K}}$ kann generell ein Funktional der Satellitenposition, -geschwindigkeit und weiteren Parametern $\boldsymbol{p}(t)$ sein. Wie schon in Abschnitt 2.3.4 festgestellt, wird in dieser Arbeit lediglich diejenige Beschleunigung berücksichtigt, die sich aus einem Schwerefeldmodell bis zu einem bestimmten Maximalgrad ergibt. Daher ist die Kräftefunktion nicht von der Satellitengeschwindigkeit abhängig. Da nur statische Gravitationsfeldmodelle verwendet werden, ist außerdem $\boldsymbol{p}$ zeitinvariant. Zur Bestimmung der Kräftefunktion ist hier also der Gradient des Gravitationspotentials zu bilden – einer Reihenentwicklung, die von den sphärischen Ortskoordinaten des Satelliten im erdfesten, geozentrischen Bezugssystem abhängt. Die Bewegungsgleichungen werden jedoch im raumfesten, geozentrischen (kartesischen) Koordinatensystem formuliert, weil man auf diese Weise keine Scheinbeschleunigungen berücksichtigen muss. Deshalb sind Transformationen zwischen den Bezugssystemen mithilfe von zeitabhängigen, nach den IERS-Richtlinien (Petit und Luzum, 2010) ermittelten Rotationsmatrizen notwendig. Somit ist $\boldsymbol{f}_{\mathrm{K}}$ vom quasi-inertialen Ortsvektor und wegen der Rotation außerdem explizit von der Zeit t abhängig, d.h. es gilt $\boldsymbol{f}_{\mathrm{K}} = \boldsymbol{f}_{\mathrm{K}}(t, \boldsymbol{X}(t), \boldsymbol{p})$.

Sämtliche Unsicherheiten hinsichtlich der Systemdynamik[1] seien stochastisch mit einem Rauschterm $\boldsymbol{w}(t)$ modelliert. Die Wahl fällt hier somit auf ein Systemmodell, das sich offenbar als eine vektorielle, zeitkontinuierliche, nicht-lineare, stochastische Differentialgleichung in der Form

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{f}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \boldsymbol{p}) + \boldsymbol{w}(t) = \begin{pmatrix} \dot{\boldsymbol{X}}(t) \\ \boldsymbol{f}_{\mathrm{K}}(t, \boldsymbol{X}(t), \boldsymbol{p}) \end{pmatrix} + \boldsymbol{w}(t), \tag{3.23}$$

darstellen lässt, worin $\boldsymbol{f} = \boldsymbol{f}(t, \boldsymbol{s}(t), \boldsymbol{p})$ die nicht-lineare Systemfunktion bezeichnet, vgl. Gl. (3.22). Da der Zustandsvektor

$$\boldsymbol{s}(t) = (X(t),\ Y(t),\ Z(t),\ \dot{X}(t),\ \dot{Y}(t),\ \dot{Z}(t))^{\top}$$

im obigen Fall sechs Freiheitsgrade aufweist, sind ebensoviele Anfangswerte vorzugeben, um zukünftige Lösungen $\boldsymbol{s}(t)$ eindeutig festzulegen. Analog zum zeitdiskreten, linearen Kalman-Filter müssen für die Filterinitialisierung neben a priori geschätzten Anfangswerten auch die zugehörigen Kovarianzen zur Verfügung stehen, d.h. $\boldsymbol{s}(0) \sim \mathcal{N}(\hat{\boldsymbol{s}}_0, \widehat{\boldsymbol{P}}_0)$.

Das Systemrauschen $\boldsymbol{w}(t) \sim \mathcal{N}(0, \boldsymbol{Q}(t))$ wird (der Einfachheit halber) wiederum als weiß, normalverteilt und erwartungswertfrei angenommen. Unter $\boldsymbol{Q}(t)$ ist die Spektraldichtematrix des Systemrauschens zu verstehen, welche *nicht* mit der Kovarianzmatrix des Systemrauschens $\boldsymbol{Q}_k$ im Zeitdiskreten übereinstimmt[2]. Auf die Modellierung weiterer Stellgrößen $\boldsymbol{u}(t)$, wie sie im Systemmodell des letzten Abschnitts noch vorgesehen waren, wird in Gleichung (3.23) verzichtet.

Der zu verdichtende Orbit ist in dieser Arbeit als Zeitreihe der Satellitenpositionen und -geschwindigkeiten mit zugehörigen Fehlermaßen gegeben. Diese Daten stellen die Hauptbeobachtungen für

1 Dazu gehören u.a. alle nicht im deterministischen Anteil berücksichtigten Störbeschleunigungen, wie z.B. die Atmosphärenreibung. Näheres zum Systemrauschen in Unterabschnitt 3.2.4.

2 Der allgemeine Zusammenhang zwischen den beiden Matrizen, die sich auch bezüglich ihrer Einheit unterscheiden, wird später noch angegeben, s. Gl. (3.45).

den Korrekturschritt des Kalman-Filters dar und sind, da sie raumfesten Bezug haben, quasi direkt mit den Zustandsgrößen verknüpft (bis auf das Messrauschen). Gleichwohl setzen wir für die Beobachtungen das allgemeinere nicht-lineare Modell

$$\boldsymbol{b}_k = \boldsymbol{h}(\boldsymbol{s}(t_k)) + \boldsymbol{v}_k, \quad k \in \mathbb{N} \tag{3.24}$$

an, um auch solche Beobachtungen auswerten zu können, die sich nur in nicht-linearen Funktionalen des Systemzustands $\boldsymbol{h}(\boldsymbol{s}(t_k))$ formulieren lassen (z.B. die K-Band-Messungen). Analog zu den Annahmen (3.5) bis (3.7) sei das Messrauschen $\boldsymbol{v}_k$ normalverteilt, mittelwertfrei, nicht mit dem Systemrauschen korreliert und habe die Kovarianzmatrix $\boldsymbol{R}_k$.

Der Prädiktionsschritt erfolgt beim gewählten Kalman-Filter also nach einem **nicht-linearen**, **zeitkontinuierlichen** Systemmodell. Die Korrektur erfolgt auf Grundlage eines Beobachtungsmodells durch Messungen, die an **diskreten** Zeitpunkten vorliegen und **nicht-linear** mit dem Systemzustand verknüpft sind. Dementsprechend wird das verwendete Filter in der Literatur häufig bezeichnet als kontinuierlich-diskretes erweitertes Kalman-Filter (engl.: „hybrid/**continuous-discrete** **extended** Kalman filter“, u.a. Gelb, 1996, S. 188; Simon, 2006, S. 403 ff.).

Das EKF basiert auf dem Ansatz, die nicht-lineare Systemfunktion $\boldsymbol{f}$ um den aktuellen Schätzwert $\bar{\boldsymbol{s}}(t)$ und die nicht-lineare Messgleichung um den prädizierten (a priori) Zustandsvektor $\bar{\boldsymbol{s}}_k$ am Messzeitpunkt zu linearisieren, um die ursprünglich für lineare Systeme ausgelegte Kalman-Filterung auch bei nicht-linearen Schätzproblemen anwenden zu können. Dieser, wie schon erwähnt, von Stanley F. Schmidt stammende Linearisierungsansatz wird im Folgenden eingehender betrachtet.

3.2.2 Linearisierung

Die mittlere Schätzung $\bar{\boldsymbol{s}}(t)$, $t \in [t_0, t_1]$ der tatsächlichen (Satelliten-) Bahn bis zur ersten Messung zum Zeitpunkt t_1 ergibt sich beim EKF entsprechend dem Systemmodell durch numerische Integration der nicht-linearen, deterministischen DGL

$$\dot{\bar{\boldsymbol{s}}}(t) = \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) \tag{3.25}$$

mit den vorab festzulegenden Anfangsbedingungen $\bar{\boldsymbol{s}}(t_0) = \hat{\boldsymbol{s}}_0$. Analog ergeben sich später weitere geschätzte Bahnen, indem dann der jeweils im Korrekturschritt bestimmte a posteriori Zustandsvektor $\hat{\boldsymbol{s}}_k$ als neuer Startwert für die Integration verwendet wird (vgl. Abb. 3.3).

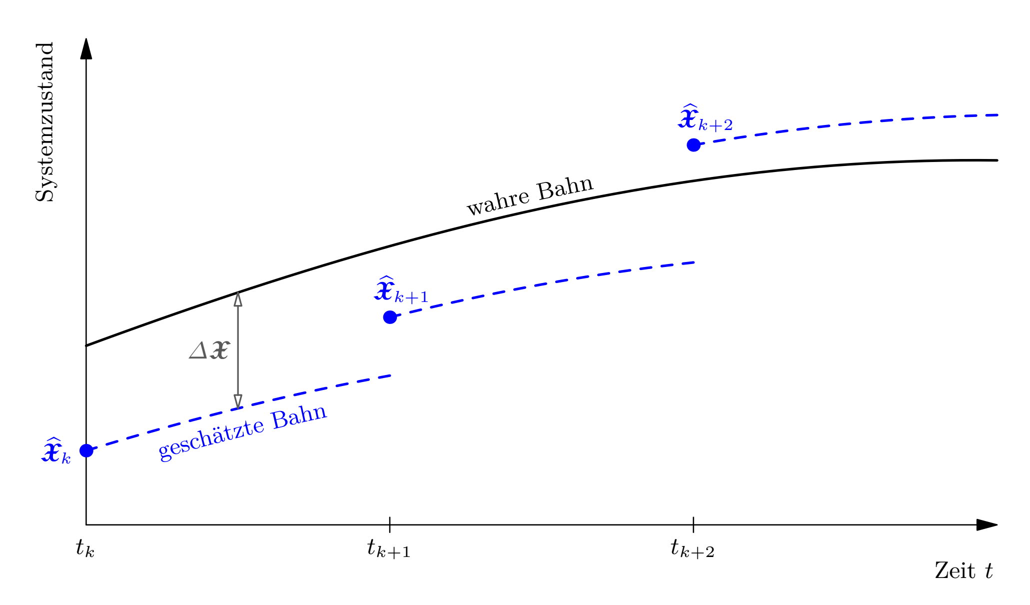


Abb. 3.3: Wahre Bahn und geschätzte Bahnen beim EKF

Definiert man $\Delta\boldsymbol{s}(t)$ als (zahlenmäßig unbekannte) Differenz zwischen der wahren Bahn und der geschätzten Bahn (vgl. Brown und Hwang, 1997, S. 336 ff.), so folgen für die Systemgleichung (3.23) und die Messgleichung (3.24) die Beziehungen

$$\dot{\bar{\boldsymbol{s}}}(t) + \Delta\dot{\boldsymbol{s}}(t) = \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t) + \Delta\boldsymbol{s}(t), \boldsymbol{p}) + \boldsymbol{w}(t) \tag{3.26}$$

$$\boldsymbol{b}_k = \boldsymbol{h}(\bar{\boldsymbol{s}}(t_k) + \Delta\boldsymbol{s}(t_k)) + \boldsymbol{v}_k. \tag{3.27}$$

Die nicht-linearen Funktionen $\boldsymbol{f}$ und $\boldsymbol{h}$ entwickelt man nun jeweils um die aktuelle Zustandsschätzung in eine Taylorreihe und bricht diese nach dem Term erster Ordnung ab:

$$\boldsymbol{f}(t, \bar{\boldsymbol{s}}(t) + \Delta\boldsymbol{s}(t), \boldsymbol{p}) = \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) + \left.\frac{\partial \boldsymbol{f}(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}(t)}\right|_{\hat{\boldsymbol{s}}(t)} [\boldsymbol{s}(t) - \bar{\boldsymbol{s}}(t)] + \mathcal{O}(\Delta\boldsymbol{s}^2) \tag{3.28}$$

$$\boldsymbol{h}(\bar{\boldsymbol{s}}(t_k) + \Delta\boldsymbol{s}(t_k)) = \boldsymbol{h}(\bar{\boldsymbol{s}}(t_k)) + \left.\frac{\partial \boldsymbol{h}(\boldsymbol{s}(t_k))}{\partial \boldsymbol{s}(t_k)}\right|_{\bar{\boldsymbol{s}}(t_k)} [\boldsymbol{s}(t) - \bar{\boldsymbol{s}}(t)] + \mathcal{O}(\Delta\boldsymbol{s}^2) \tag{3.29}$$

Die darin auftretenden partiellen Ableitungen

$$\boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) := \left.\frac{\partial \boldsymbol{f}(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}(t)}\right|_{\bar{\boldsymbol{s}}(t)} = \begin{pmatrix} \left.\frac{\partial f_1(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}_1(t)}\right|_{\bar{\boldsymbol{s}}(t)} & \cdots & \left.\frac{\partial f_1(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}_n(t)}\right|_{\bar{\boldsymbol{s}}(t)} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial f_n(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}_1(t)}\right|_{\bar{\boldsymbol{s}}(t)} & \cdots & \left.\frac{\partial f_n(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}_n(t)}\right|_{\bar{\boldsymbol{s}}(t)} \end{pmatrix}_{(n\times n)} \tag{3.30}$$

$$\boldsymbol{H}(\bar{\boldsymbol{s}}(t_k)) := \left.\frac{\partial \boldsymbol{h}(\boldsymbol{s}(t_k))}{\partial \boldsymbol{s}(t_k)}\right|_{\bar{\boldsymbol{s}}(t_k)} = \begin{pmatrix} \left.\frac{\partial h_1(\boldsymbol{s}(t_k))}{\partial \boldsymbol{s}_1(t_k)}\right|_{\bar{\boldsymbol{s}}(t_k)} & \cdots & \left.\frac{\partial h_1(\boldsymbol{s}(t_k))}{\partial \boldsymbol{s}_n(t_k)}\right|_{\bar{\boldsymbol{s}}(t_k)} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial h_m(\boldsymbol{s}(t_k))}{\partial \boldsymbol{s}_1(t_k)}\right|_{\bar{\boldsymbol{s}}(t_k)} & \cdots & \left.\frac{\partial h_m(\boldsymbol{s}(t_k))}{\partial \boldsymbol{s}_m(t_k)}\right|_{\bar{\boldsymbol{s}}(t_k)} \end{pmatrix}_{(m\times n)} \tag{3.31}$$

sind die Jacobi-Matrizen der Funktionen $\boldsymbol{f}$ und $\boldsymbol{h}$. Deren Dimensionen werden durch die Zahl n der Elemente des Zustandsvektors und die Anzahl der Einzelbeobachtungen m im Messvektor bestimmt. Man erhält damit das lineare Fehlersystemmodell

$$\dot{\bar{\boldsymbol{s}}}(t) + \Delta\dot{\boldsymbol{s}}(t) \approx \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) + \boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})\Delta\boldsymbol{s}(t) + \boldsymbol{w}(t) \tag{3.32}$$

$$\xRightarrow{(3.25)} \Delta\dot{\boldsymbol{s}}(t) \approx \boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})\Delta\boldsymbol{s}(t) + \boldsymbol{w}(t) \tag{3.33}$$

zur Prädiktion der Schätzfehler[1] sowie die linearisierte Innovationsgleichung

$$\boldsymbol{b}_k - \boldsymbol{h}(\bar{\boldsymbol{s}}(t_k)) \approx \boldsymbol{H}(\bar{\boldsymbol{s}}(t_k))\Delta\boldsymbol{s}(t_k) + \boldsymbol{v}_k. \tag{3.34}$$

Unterschiede zwischen erweiterter und linearisierter Kalman-Filterung

Die beiden obigen linearen Modelle (Gl. (3.33) und (3.34)) werden beim EKF genutzt, um das nicht-lineare Systemmodell bzw. das nicht-lineare Beobachtungsmodell in der Umgebung des aktuell geschätzten Systemzustands zu approximieren. Generell kann die Approximation nur dann gut gelingen, wenn die Abweichung zwischen der wahren Bahn und der geschätzten Trajektorie bzw. die Differenz zwischen tatsächlicher Beobachtung und erwarteter Beobachtung (Innovation) genügend

1 Die absoluten Abweichungen $\Delta\boldsymbol{s}$ zur wahren Bahn sind zwar unbekannt, jedoch sind die Schätzfehlerkovarianzen des Anfangszustands bekannt, die später auf Grundlage dieses linearen Modells prädiziert werden.

klein ist. Bei einem anderen auf dem Linearisierungsprinzip basierenden Filter, dem linearisierten Kalman-Filter (LKF), ist dieser Aspekt problematischer, weil hier die Linearisierung nicht wie beim EKF um den aktuellen Schätzwert, sondern um eine vorausberechnete Trajektorie einer nominalen Bewegung erfolgt. Die geschätzte Bahn wird beim LKF also nicht nach jeder Beobachtung auf einen korrigierten Systemzustand zurückgesetzt, sondern es bleibt zunächst *eine* geschätzte Bahn erhalten und beim Korrekturschritt wird lediglich eine neue Verbesserung zum nominellen Zustandsvektor geschätzt, die jedoch nicht in die nächste Prädiktion des Zustandsvektors selbst eingeht. Stattdessen wird diese Verbesserung linear mithilfe der Transitionsmatrix prädiziert und bei der nächsten Korrektur wieder angepasst usw. (zum genauen Ablauf des Algorithmus siehe z.B. Vallado und McClain, 2007, S. 784). Beim LKF genügt es somit, *ein* AWP – das der nominellen geschätzten Bahn – zu lösen, weshalb der Rechenaufwand erheblich geringer als beim EKF ausfällt, zumal auch die Transitions- und Verstärkungsmatrizen bereits ‚off-line' berechnet werden können, d.h. bevor überhaupt Messungen anfallen. Daher arbeitet das LKF besonders bei Echtzeitanwendungen sehr effektiv. Andererseits führt dieses Konzept unmittelbar dazu, dass die Abweichung zwischen den beiden Bahnen mit der Zeit tendenziell immer weiter zunimmt. Mit der ansteigenden Differenz gewinnen aber gerade die bei der Taylorentwicklung vernachlässigten Terme höherer Ordnung an Bedeutung. Die Referenzbahn sollte demzufolge bereits sehr genau der wahren Bahn entsprechen, denn ansonsten könnten die ansteigenden Abweichungen zu fehlerhaften Schätzwerten und letztlich zu einer Divergenz des Filters führen. Falls die geschätzte Verbesserung zur nominellen Bahn zu groß wird, kann ggf. das Filter ‚zurückgesetzt' werden, indem man eine neue nominelle Bahn berechnet, die z.B. beim aktuell geschätzten Systemzustand beginnt. Das EKF gilt bei nicht-linearen Problemen als robuster und vermag höhere Schätzgenauigkeiten zu liefern. Die Vorausberechnung der Systemmatrizen ist indes aufgrund deren Abhängigkeit von den ‚Echtzeit'-Zustandsschätzwerten nicht möglich. Der Nachteil des daraus resultierenden höheren Rechenaufwands ist bei der Anwendung der Orbitverdichtung weniger kritisch zu sehen, da es sich ohnehin um eine Aufgabe der nachträglichen Datenauswertung („postprocessing") handelt. Daher wurde hier für die Orbitverdichtung dem EKF der Vorzug gegeben.

Sowohl beim linearisierten als auch beim erweiterten Kalman-Filter ist wegen der Taylorreihenentwicklung zu fordern, dass die Systemfunktionen $\boldsymbol{f}$ und $\boldsymbol{h}$ zumindest zweimal stetig differenzierbar sind. Beide Filter gelten als Suboptimalfilter, da die Schätzung aufgrund der Approximationsfehler nicht mehr optimal im Sinne der Bayes'schen Schätztheorie ist (z.B. Grewal und Andrews, 2008, S. 383 ff.). Brauchbare Ergebnisse erhält man mit den Filtern dennoch; nämlich insbesondere dann, wenn sich das Problem ‚quasilinear' verhält, sodass die durch die lineare Näherung resultierenden Fehler die Genauigkeiten der Zustandsschätzung nur in tolerierbarem Maße beeinflussen. Dies ist der Fall, wenn die Approximationsfehler insignifikant gegenüber dem Mess- und Systemrauschen sind. Grewal und Andrews (2008, S. 297) empfehlen daher einen Test auf die Quasi-Linearität des Problems. Dieser spezielle Test wird hier ausgelassen, da bereits zahlreiche erfolgreiche Orbitbestimmungen für LEO-Satelliten mithilfe des EKF durchgeführt wurden (in Vallado und McClain, 2007, S. 788 wird auf viele dieser Anwendungen verwiesen).

3.2.3 Filtergleichungen

In Tafel 3.2 sind die Gleichungen des verwendeten erweiterten Kalman-Filters zusammengestellt. Im Prädiktionsschritt werden ausschließlich die Fehlerkovarianzen nach einem linearen Modell fortgeschrieben (s.u.); da man die Trajektorie möglichst exakt vorhersagen möchte, wird der Systemzustand $\hat{\boldsymbol{s}}_k$ hingegen gemäß des *nicht-linearen* Systemmodells (3.25) prädiziert. Ein zukünftiger Systemzustand zum Zeitpunkt t ist demnach formal mit

$$\bar{\boldsymbol{s}}(t) = \hat{\boldsymbol{s}}_k + \int_{t_k}^{t} \boldsymbol{f}(\tau, \bar{\boldsymbol{s}}(\tau), \boldsymbol{p})\, \mathrm{d}\tau, \quad t \in [t_k, t_{k+1}] \tag{3.35}$$

angebbar. Stark vereinfachte Näherungen des Integralterms, z.B. mithilfe des zentralen Differenzenquotienten, werden aufgrund der hohen Genauigkeitsansprüche nicht in Betracht gezogen. Stattdessen nutzt man in dieser Arbeit zur Lösung von DGLen, wie später noch eingehend beschrieben wird, ein MATLAB-eigenes numerisches Integrationsverfahren.

Tafel 3.2: Gleichungen des kontinuierlich-diskreten erweiterten Kalman-Filters

Schritt 1 – Prädiktion

- Voraussage des Systemzustands $\bar{\boldsymbol{s}}_{k+1}$:

  gemäß Gl. (3.25) durch numerische Integration des nicht-linearen AWPs

  $$\dot{\bar{\boldsymbol{s}}}(t) = \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}), \quad \bar{\boldsymbol{s}}(t_k) = \hat{\boldsymbol{s}}_k$$

  zwischen zwei Beobachtungs-/Korrekturzeitpunkten $t \in [t_k, t_{k+1}]$ in der gewünschten zeitdiskreten Auflösung δt. Der erste Startwert $\hat{\boldsymbol{s}}_0$ muss vorab bestimmt werden.

- Voraussage der Schätzfehlerkovarianzmatrix $\overline{\boldsymbol{P}}_{k+1}$:

  – Variante 1: numerische Integration der Matrix-DGL

  $$\dot{\overline{\boldsymbol{P}}}(t) = \boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})\overline{\boldsymbol{P}}(t) + \overline{\boldsymbol{P}}(t)\boldsymbol{F}^\top(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) + \boldsymbol{Q}(t) \tag{3.36}$$

  beginnend bei $\overline{\boldsymbol{P}}(t_k) = \widehat{\boldsymbol{P}}_k$ bzw. $\widehat{\boldsymbol{P}}_0$ mit der Jacobi-Matrix

  $$\boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) = \left.\frac{\partial \boldsymbol{f}(t, \boldsymbol{s}(t), \boldsymbol{p})}{\partial \boldsymbol{s}(t)}\right|_{\bar{\boldsymbol{s}}(t)}$$

  gemäß (3.30).

  – Variante 2: löse das Matrix-AWP (3.44) für $\boldsymbol{\Phi}_k$ und berechne $\overline{\boldsymbol{P}}_{k+1}$ nach

  $$\overline{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k \widehat{\boldsymbol{P}}_k \boldsymbol{\Phi}_k^\top + \boldsymbol{Q}_k \tag{3.37}$$

Schritt 2 – Korrektur durch Messungen

Innovation:

$$\boldsymbol{d}_{k+1} = \boldsymbol{b}_{k+1} - \boldsymbol{h}(\bar{\boldsymbol{s}}_{k+1}) \tag{3.38}$$

Kalman-Verstärkungsmatrix analog zu Gl. (3.16)

$$\boldsymbol{K}_{k+1} = \overline{\boldsymbol{P}}_{k+1}\boldsymbol{H}_{k+1}^\top \left[\boldsymbol{H}_{k+1}\overline{\boldsymbol{P}}_{k+1}\boldsymbol{H}_{k+1}^\top + \boldsymbol{R}_{k+1}\right]^{-1}$$

mit der Jacobi-Matrix

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}(\bar{\boldsymbol{s}}(t_{k+1})) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{s}(t_{k+1}))}{\partial \boldsymbol{s}(t_{k+1})}\right|_{\bar{\boldsymbol{s}}(t_{k+1})}$$

gemäß (3.31).

Korrigierter Systemzustand und Kovarianzmatrix (analog zu Gl. (3.17) und (3.18)):

$$\hat{\boldsymbol{s}}_{k+1} = \bar{\boldsymbol{s}}_{k+1} + \boldsymbol{K}_{k+1}\boldsymbol{d}_{k+1} = \bar{\boldsymbol{s}}_{k+1} + \boldsymbol{\delta}\hat{\boldsymbol{s}}_{k+1}$$

$$\widehat{\boldsymbol{P}}_{k+1} = \overline{\boldsymbol{P}}_{k+1} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1}\overline{\boldsymbol{P}}_{k+1}$$

Um die Fehlerkovarianzen zu prädizieren, werden hier zwei Varianten umgesetzt. Die erste ist an das zeitkontinuierliche Kalman-Bucy-Filter, die zweite an das zeitdiskrete Kalman-Filter angelehnt.

In der ersten Variante schreibt man die Kovarianzen durch numerische Integration der ‚reduzierten' Riccati[1]-DGL (3.36) linear zwischen den Integrationszeitpunkten fort. Üblicherweise ist in der Riccati-DGL, wie sie beim Kalman-Bucy-Filter zum Einsatz kommt, noch ein Term zur Verarbeitung der Messungen enthalten. Dieser entfällt hier jedoch beim kontinuierlich-diskreten EKF, da die Messungen an diskreten Zeitpunkten anfallen und demzufolge in einem separaten Korrekturschritt verarbeitet werden. Herleitungen der Riccati-DGL findet man beispielsweise in Gelb (1996, S. 121-122), Simon (2006, S. 233 ff. bzw. 404) und Maybeck (1979, S. 163 ff., Gl. 4-120). Die Systemmatrix $\boldsymbol{F}$ ist in Gl. (3.36) gemäß der Linearisierung aus Gleichung (3.30) zu berechnen. $\boldsymbol{F}$ wird auch als Dynamikmatrix bezeichnet, da sie beschreibt, wie sich die modellierten Kräfte mit der Zeit ändern. Hier ist $\boldsymbol{F}$ von der Zeit, den Schwerefeldparametern $\boldsymbol{p}$ und dem aktuell prädizierten Zustandsvektor abhängig, der sich aber gerade bei jedem Integrationszeitpunkt ändert. Das bedeutet, dass die Riccati-DGL simultan mit der nicht-linearen Systemgleichung zu integrieren ist.

Neben der beschriebenen, beim gewählten kontinuierlich-diskreten EKF gängigen ersten Variante lassen sich die Kovarianzen der Schätzfehler auch mithilfe der vom herkömmlichen Kalman-Filter bekannten zeitdiskreten Vorschrift (3.37) prädizieren. Diese zweite Variante wurde vor allem deshalb gewählt, weil hierbei die in der Satellitengeodäsie bekannten Variationsgleichungen zu lösen sind, um die Transitionsmatrix $\boldsymbol{\Phi}_k$ in Gl. (3.37) zu bestimmen. Dazu geht man zunächst von folgender Lösung der DGL (3.33) aus (vgl. Vallado und McClain, 2007, S. 772):

$$\Delta\boldsymbol{s}_{k+1} \approx \boldsymbol{\Phi}_k \Delta\boldsymbol{s}_k + \boldsymbol{w}_k. \tag{3.39}$$

Diese Approximation erlaubt die lineare, zeitdiskrete Prädiktion des Schätzfehlervektors $\Delta\boldsymbol{s}_k$ mithilfe der Transitionsmatrix $\boldsymbol{\Phi}$. Offenbar weist Gl. (3.39) dieselbe Struktur auf wie die Systemgleichung der linearen Kalman-Filterung. Für die Transitionsmatrix gilt dabei entsprechend Gleichung (3.12)

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \left.\frac{\partial \bar{\boldsymbol{s}}(t)}{\partial \hat{\boldsymbol{s}}_k}\right|_{t=t_{k+1}}. \tag{3.40}$$

Wegen der Akzentzeichen könnte man zunächst meinen, dass hier keine Abhängigkeit zwischen den beiden Zustandsvariablen vorliegt. Im Allgemeinen ist $\boldsymbol{\Phi}_k$ aber keine Nullmatrix, denn zukünftige Systemzustände sind zufolge des dynamischen Modells vom Anfangszustand abhängig. Am Startzeitpunkt ist $\boldsymbol{\Phi}(t_k, t_k)$ identisch mit der Einheitsmatrix, weil der a posteriori Systemzustand der vorangegangenen Korrektur $\hat{\boldsymbol{s}}_k$ mit dem prädizierten Systemzustand am Ausgangszeitpunkt t_k übereinstimmt ($\hat{\boldsymbol{s}}_k = \bar{\boldsymbol{s}}(t_k)$). Im Fall nicht-linearer Problemstellungen mit hohen Genauigkeitsanforderungen kann die Transitionsmatrix für spätere Zeitpunkte meist nur durch numerische Integration eines Matrix-AWPs bestimmt werden. Dieses erhält man, indem man zunächst die nicht-lineare Systemgleichung (3.25)

$$\dot{\bar{\boldsymbol{s}}}(t) = \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})$$

nach dem Ausgangszustand differenziert

$$\frac{\partial \dot{\bar{\boldsymbol{s}}}(t)}{\partial \hat{\boldsymbol{s}}_k} = \frac{\partial}{\partial \hat{\boldsymbol{s}}_k} \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}), \tag{3.41}$$

anschließend auf der linken Seite die Reihenfolge der Ableitungen tauscht

$$\frac{\partial \dot{\bar{\boldsymbol{s}}}(t)}{\partial \hat{\boldsymbol{s}}_k} = \frac{\partial}{\partial \hat{\boldsymbol{s}}_k}\left(\frac{\mathrm{d}}{\mathrm{d}t}\bar{\boldsymbol{s}}(t)\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \bar{\boldsymbol{s}}(t)}{\partial \hat{\boldsymbol{s}}_k}\right) \tag{3.42}$$

1 Benannt nach dem venezianischen Grafen und Privatgelehrten Jacopo Francesco Riccati (1676-1754), s. z.B. Wirsching (2006, S. 15).

und schließlich die rechte Seite entsprechend der Kettenregel um $\partial\bar{\boldsymbol{s}}(t)$ erweitert (vgl. Vallado und McClain, 2007, S. 794):

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\bar{\boldsymbol{s}}(t)}{\partial\hat{\boldsymbol{s}}_k}\right) = \left.\frac{\partial\boldsymbol{f}(t,\boldsymbol{s}(t),\boldsymbol{p})}{\partial\boldsymbol{s}(t)}\right|_{\bar{s}(t)} \frac{\partial\bar{\boldsymbol{s}}(t)}{\partial\hat{\boldsymbol{s}}_k} \tag{3.43}$$

Abgekürzt mit den bereits definierten Systemmatrizen und der oben genannten Anfangsbedingung für die Transitionsmatrix erhält man das Matrix-AWP

$$\dot{\boldsymbol{\Phi}}(t,t_k) = \boldsymbol{F}(t,\bar{\boldsymbol{s}}(t),\boldsymbol{p})\boldsymbol{\Phi}(t,t_k), \quad \boldsymbol{\Phi}(t_k,t_k) = \mathbf{I}. \tag{3.44}$$

Diese Differentialgleichungen, welche die partiellen Ableitungen nach dem Startvektor (d.h. die Transitionsmatrix) als unbekannte, zu bestimmende Funktionen enthalten, sind in der Satellitengeodäsie unter dem Begriff der *Variationsgleichungen* bekannt (Ballani, 1988, S. 193). Im Falle eines n-dimensionalen Zustandsvektors müssen hierbei, da $\boldsymbol{\Phi}$ die Dimension $n \times n$ hat, n^2 DGLen erster Ordnung (numerisch) bis zum jeweils nächsten Messzeitpunkt $t = t_{k+1}$ gelöst werden (vgl. Schrick, 1977, S. 48). Im Gegensatz zum linearen Kalman-Filter ist wird die Transitionsmatrix beim EKF nicht mehr für die Prädiktion der Systemzustände verwendet, sondern ausschließlich für die Fortschreibung der Schätzfehlerkovarianzen. Durch Bildung des Erwartungswertes von Gl. (3.39) folgt die zur Prädiktion der Schätzfehlerkovarianzmatrix genutzte Gleichung (3.37):

$$\overline{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k\widehat{\boldsymbol{P}}_k\boldsymbol{\Phi}_k^\top + \boldsymbol{Q}_k.$$

Hierin ist der erste Term für die (lineare) Fortpflanzung der Schätzfehlerkovarianzen des Startpunkts bis zum nächsten Messpunkt zufolge der modellierten Systemdynamik zuständig, denn die Transitionsmatrix ergibt sich ja gerade aus der Jacobi-Matrix $\boldsymbol{F}$ des verwendeten Kräftemodells. Im zweiten Term, der hier im Zeitdiskreten die Kovarianzmatrix $\boldsymbol{Q}_k$ des Systemrauschens $\boldsymbol{w}_k$ ist, soll den Unsicherheiten im ersten Term und weiteren, darüberhinaus bestehenden Fehlereinflüssen Rechnung getragen werden.

Anhand einer numerischen Untersuchung wird später noch nachgewiesen, dass die zur Fortpflanzung der Kovarianzmatrix $\widehat{\boldsymbol{P}}_k$ eingeführte Linearisierung in dieser Anwendung zulässig bzw. unkritisch ist. Grundsätzlich ist es möglich, mit beiden Varianten zur Prädiktion der Kovarianzmatrizen identische Ergebnisse zu erhalten; hier wird jedoch bei der zweiten Variante das Systemrauschen auf andere Weise modelliert als bei der ersten Variante. Darauf und auf die Bestimmung der Matrizen $\boldsymbol{Q}_k$ und $\boldsymbol{Q}(t)$ wird im nächsten Abschnitt eingegangen.

Nachdem der Zustandsvektor und die Kovarianzmatrix entweder nach der ersten oder der zweiten Variante bis zum nächsten Beobachtungszeitpunkt vorausberechnet wurde, ist der Korrekturschritt des EKFs durchzuführen. Zuerst bestimmt man gemäß Gleichung (3.38) die Innovation, d.h. die Residuen zwischen den tatsächlichen Beobachtungen und den vorausberechneten Beobachtungen, wobei letztere aus dem nicht-linearen Zusammenhang $\boldsymbol{h}(\bar{\boldsymbol{s}}_{k+1})$ folgen. Die übrige Vorgehensweise des Korrekturschritts geht dank des linearisierten Beobachtungsmodells (3.34) analog zu den vom linearen Kalman-Filter bekannten Gleichungen vonstatten. Unter Zuhilfenahme der linearen Näherung $\boldsymbol{H}_{k+1}$ ergibt sich die Kalman-Verstärkung gemäß der bekannten Gleichung (3.16). Die differentielle Korrektur $\boldsymbol{K}_{k+1}\boldsymbol{d}_{k+1}$ addiert man wie gehabt als Zuschlag zum prädizierten Systemzustand. Zuletzt ist noch die Schätzfehlerkovarianzmatrix zu verbessern, vorzugsweise wieder mithilfe einer numerisch stabileren Alternative, hier der Joseph-Form, wobei ebenfalls die approximierte Messmatrix $\boldsymbol{H}_{k+1}$ beteiligt ist.

3.2.4 Systemrauschen und Filterstabilität

Das Systemrauschen ist im zeitkontinuierlichen Fall mit der Spektraldichtematrix $\boldsymbol{Q}(t)$ (Gl. (3.36)) und im Zeitdiskreten mit der Kovarianzmatrix $\boldsymbol{Q}_k$ (Gl. (3.37)) modelliert. In beiden Fällen kommt den Matrizen die Aufgabe zu, sämtliche Unsicherheiten hinsichtlich der Systemdynamik abzubilden. Dazu gehören insbesondere (s. z.B. Boden, 1986, S. 29):

- Ungenauigkeiten durch die lineare Approximation der nicht-linearen Modelle
- Fehler durch nicht modellierte Einflüsse auf die Systemdynamik; hier also sämtliche nicht bei der Orbitintegration berücksichtigten Störbeschleunigungen, die z.B. zustande kommen durch Drittkörper (Sonne, Mond), vernachlässigte Terme höherer Ordnung des Gravitationspotentials, Atmosphärenreibung, Albedo-Effekte etc.
- Unsicherheiten im verwendeten nicht-linearen, deterministischen Systemmodell, hier etwa die Unsicherheiten der berücksichtigten Koeffizienten des Gravitationspotentials.

Aufgrund der Ursachen dieser Unsicherheiten müsste die Kovarianzfunktion des Systemrauschens eigentlich durch Fehlermodelle beschrieben werden, die auf der Physik des jeweiligen Problems beruhen. Für die Problemstellung der Orbitbestimmung eines Satelliten hat bereits James R. Wright einen solchen physikalischen Ansatz entwickelt (vgl. Wright, 1981; Wright et al., 2008). Sein Konzept sieht vor, in der zeitdiskreten Prädiktionsgleichung (3.37) die Kovarianzmatrix des Systemrauschens $\boldsymbol{Q}_k$ durch eine symmetrische Matrix $\boldsymbol{P}^{\iint}_{k+1,k}$ zu ersetzen, wobei der Integrand des Doppelintegrals eine komplizierte Funktion unterschiedlicher sphärisch-harmonischer Parameter ist, u.a. der Gradvarianzen. Dabei kann man sich folgendes Prinzip vorstellen: verwendet man beispielsweise ein Gravitationsfeld bis Grad und Ordnung 10 für das deterministische Systemmodell, müssen mit $\boldsymbol{P}^{\iint}_{k+1,k}$ zum einen diejenigen Fehler abgeschätzt werden, die durch alle vernachlässigten Entwicklungsgrade > 10 zustande kommen und zum anderen die Unsicherheiten der verwendeten Potentialkoeffizienten selbst. In Wright (1981) ist aufgezeigt, dass die Autokorrelationsfunktion der nach diesem Ansatz berechneten Fehlerkovarianzen *nicht* mit der Dirac-Funktion (Autokorrelationsfunktion von weißem Rauschen) übereinstimmt, weshalb die häufig beim Kalman-Filter getroffene Annahme weißen Rauschens streng betrachtet nicht gültig ist. Vor allem im Fall langer Prädiktionszeiten (in der Untersuchung von Wright, 1981, S. 309: 100 Min.) ist es sicherlich sinnvoll, die in Wright et al. (2008) beschriebene Matrix $\boldsymbol{P}^{\iint}_{k+1,k}$ als Kovarianzmatrix $\boldsymbol{Q}_k$ zu verwenden. Dennoch werden durch diesen Ansatz auch ‚nur' die Fehlerkovarianzen des Gravitationsfeldmodells („gravity error covariances") im Systemrauschen berücksichtigt, d.h. die anderen oben genannten Fehlerquellen werden dadurch nicht abgeschätzt.

Da die Berechnung der Matrix $\boldsymbol{P}^{\iint}_{k+1,k}$ sehr aufwendig (und dementsprechend schwer implementierbar) wäre, und zudem hier nur vergleichsweise kurze Prädiktionen von 60 Sekunden durchzuführen sind, wird in dieser Arbeit nur das einfache, weiße, normalverteilte Rauschen umgesetzt. Dies bedeutet, dass keine Korrelationen zwischen den Prädiktionsfehlern (d.h. den Komponenten des Systemrauschens $\boldsymbol{w}(t)$) bestehen, die Matrizen $\boldsymbol{Q}_k$ bzw. $\boldsymbol{Q}(t)$ also Diagonalstruktur besitzen.

Bei der zeitkontinuierlichen Prädiktion der Kovarianzen mithilfe der Riccati-DGL (3.36) (Variante 1) ist die Spektraldichtematrix $\boldsymbol{Q}(t)$ in die Integration miteinbezogen und hat wegen des weißen Rauschens nicht nur Diagonalstruktur sondern auch konstante Einträge[1], d.h. $\boldsymbol{Q}(t) = \boldsymbol{Q}$.

Die in der zweiten Variante verwendete Kovarianzmatrix des Systemrauschens $\boldsymbol{Q}_k$ ist nicht gleichbedeutend mit der Spektraldichtematrix[2] und hat daher nicht automatisch ebenfalls zeitlich konstante

1 Weißes Rauschen hat eine *konstante* spektrale Leistungsdichte.

2 Anhand Gl. (3.36) und (3.37) sieht man sofort, dass sich die Einheiten beider Matrizen unterscheiden: $\boldsymbol{Q}$ hat die gleichen Einheiten wie $\dot{\boldsymbol{P}}$ und $\boldsymbol{Q}_k$ die gleichen wie $\boldsymbol{P}$.

Einträge. Den Zusammenhang zwischen den beiden Matrizen geben unter anderem Schrick (1977, S. 108) und Gelb (1996, S. 293) mit der Integralgleichung

$$\boldsymbol{Q}_k = \int\limits_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau)\boldsymbol{Q}(\tau)\boldsymbol{\Phi}^\top(t_{k+1}, \tau)\, \mathrm{d}\tau \tag{3.45}$$

an, welche ihrerseits eine Lösung der bilinearen Matrix-DGL

$$\dot{\boldsymbol{Q}}_k(t) = \boldsymbol{F}(t, \bar{\boldsymbol{s}}(t))\boldsymbol{Q}_k(t) + \boldsymbol{Q}_k(t)\boldsymbol{F}^\top(t, \bar{\boldsymbol{s}}(t)) + \boldsymbol{Q}(t), \quad \boldsymbol{Q}_k(t_k) = \boldsymbol{0} \tag{3.46}$$

zum Zeitpunkt $t = t_{k+1}$ ist[1]. Da in unserem Fall $\boldsymbol{F}$ und damit $\boldsymbol{\Phi}$ zeitvariabel sind, müsste zur Bestimmung der Kovarianzmatrix aus einer Spektraldichtematrix dieses AWP gelöst werden. Dann wäre die zweite Variante zur ersten Variante äquivalent. In der zweiten Variante müssten hierbei für einen sechsdimensionalen Zustandsvektor insgesamt $6 + 36 + 36 = 78$ DGLen erster Ordnung numerisch integriert werden. Dies würde auf einem momentan handelsüblichen PC entweder lange Rechenzeiten zur Folge haben oder man müsste die Anzahl der berücksichtigten Kräfte im Kräftemodell deutlich limitieren, was beides nachteilig wäre.

Daher wird darauf verzichtet, das AWP (3.46) numerisch zu integrieren. Stattdessen wird für die zweite Variante eine Näherungslösung hergeleitet. Dazu setzt man *nur hier* zur Fortschreibung des Systemrauschens ein stückweise stationäres (zeitunabhängiges) Systemverhalten zwischen zwei Messzeitpunkten t_k und t_{k+1} voraus. In diesem Zeitabschnitt gilt dann für die Jacobi-Matrix $\boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) = \boldsymbol{F}(t_k, \hat{\boldsymbol{s}}_k, \boldsymbol{p}) = \text{const.}$, sodass sich die Transitionsmatrix geschlossen unter Verwendung der Matrixexponentialfunktion zu

$$\boldsymbol{\Phi}(t, t_k) = \exp\left(\boldsymbol{F} \cdot (t - t_k)\right) = \sum_{\ell=0}^{\infty} \frac{\boldsymbol{F}^\ell}{\ell\,!}(t - t_k)^\ell, \quad t \in [t_k, t_{k+1}] \tag{3.47}$$

darstellen lässt Schrick (1977, S. 314). Bricht man diese unendliche Reihendarstellung nach dem Term erster Ordnung ab (lineare Euler-Approximation), erhält man den Zusammenhang

$$\boldsymbol{\Phi}(t, t_k) \approx \mathbf{I} + (t - t_k)\boldsymbol{F}(\hat{\boldsymbol{s}}_k) \tag{3.48}$$

als lineare Näherung für die Transitionsmatrix. Da die Spektraldichtematrix $\boldsymbol{Q}$ konstant im betrachteten Zeitabschnitt $\Delta t = t_{k+1} - t_k$ ist, kann Gleichung (3.45) nach Einsetzen der Näherung (3.48) wie folgt analytisch integriert werden (vgl. Schlotterer, 2008, S. 29):

$$\begin{aligned}\boldsymbol{Q}_k &= \int\limits_0^{\Delta t} \boldsymbol{\Phi}(\tau)\boldsymbol{Q}\boldsymbol{\Phi}^\top(\tau)\, \mathrm{d}\tau \approx \int\limits_0^{\Delta t} (\mathbf{I} + \tau\boldsymbol{F})\boldsymbol{Q}(\mathbf{I} + \tau\boldsymbol{F})^\top\, \mathrm{d}\tau = \int\limits_0^{\Delta t} (\mathbf{I} + \tau\boldsymbol{F})(\boldsymbol{Q} + \tau\boldsymbol{Q}\boldsymbol{F}^\top)\, \mathrm{d}\tau \\ &= \int\limits_0^{\Delta t} \boldsymbol{Q} + \tau\boldsymbol{Q}\boldsymbol{F}^\top + \tau\boldsymbol{F}\boldsymbol{Q} + \tau^2\boldsymbol{F}\boldsymbol{Q}\boldsymbol{F}^\top\, \mathrm{d}\tau = \Delta t\boldsymbol{Q} + \frac{1}{2}\Delta t^2(\boldsymbol{Q}\boldsymbol{F}^\top + \boldsymbol{F}\boldsymbol{Q}) + \frac{1}{3}\Delta t^3\boldsymbol{F}\boldsymbol{Q}\boldsymbol{F}^\top\end{aligned} \tag{3.49}$$

In dieser Anwendung sind in der Regel die Einträge der Jacobi-Matrix $\boldsymbol{F}_{[ij]} \leq 1$ sowie $\boldsymbol{Q}_{[ij]} \ll 1$. Da hier zudem die Zeitintervalle Δt kurz sind, wird für $\boldsymbol{Q}_k$ der erste, in Δt lineare Term

$$\boldsymbol{Q}_k \approx \Delta t \cdot \boldsymbol{Q} \tag{3.50}$$

als (grobe) Näherung für das AWP (3.46) verwendet. Bei der Orbitverdichtung ist die Matrix des Systemrauschens an weiteren Zeitpunkten $i \cdot \delta t$ zwischen den beiden Messzeitpunkten zu bestimmen,

1 Generell kann die Spektraldichtematrix durch Multiplikation mit der Dirac-Funktion (Einheit $^1/_\mathrm{s}$ = Hz) in die zugehörige Kovarianzmatrix überführt werden.

die das Intervall Δt in m gleiche Zeitabschnitte δt von z.B. einer Sekunde unterteilen ($\Delta t = m \cdot \delta t$). Angewandt auf Gleichung (3.50) folgt

$$\boldsymbol{Q}_k(i \cdot \delta t) \approx i \cdot \delta t \cdot \boldsymbol{Q}, \quad i = 0, \ldots, m. \tag{3.51}$$

Diese Vereinfachung erhält man auch, wenn man die in Gelb (1996, S. 298) angegebene Reihenentwicklung für $\boldsymbol{Q}_k$ nach dem linearen Term abbricht. Daran ist sofort ersichtlich, dass nach jedem Prädiktionsschritt (der ‚Länge' δt) die aufaddierte $\boldsymbol{Q}_k$-Matrix größer wird.

Hierbei und bei der ersten Variante wird also unterstellt, dass die oben genannten Unsicherheiten, die das Systemrauschen ausmachen, mit voranschreitender Integrationszeit ansteigen. Auf die Approximationsfehler und die Unsicherheiten des verwendeten Kräftemodells mag dies zutreffen. Zumindest bei langen Integrationszeiten ist dies mit Blick auf diejenigen Fehler, die durch die im dynamischen Modell vernachlässigten Störbeschleunigungen entstehen, aber keineswegs sicher, da sich diese Beschleunigungen innerhalb der Integrationszeit ändern können.

Im Falle langer Integrationszeiten kann es deshalb angemessener sein, einfach auf alle mit der Transitionsmatrix prädizierten Kovarianzen einen bestimmten konstanten Zuschlag für das Systemrauschen zu addieren, d.h. man betrachtet die Kovarianzmatrix $\boldsymbol{Q}_k$ auch im Zeitdiskreten als zeitinvariant, wählt konstante Werte auf der Hauptdiagonalen und addiert diese Matrix gleichermaßen auf alle prädizierten Kovarianzmatrizen $\boldsymbol{\Phi}_k \widehat{\boldsymbol{P}}_k \boldsymbol{\Phi}_k^\top$. Dann spiegelt sozusagen die Matrix des Systemrauschens in jedem Integrationsschritt die angenommene Summe der Unsicherheiten wider.

3.2.4.1 Gewählte Methoden zur Prädiktion der Schätzfehlerkovarianzen

Folgende Methoden wurden umgesetzt, um die Schätzfehlerkovarianzmatrix $\widehat{\boldsymbol{P}}_k$ entsprechend den zugrundeliegenden linearisierten Fehlersystemmodellen (3.33) bzw. (3.39) zu prädizieren:

- Methode `'ric'`*: Numerische Integration der Riccati-DGL (3.36) mit konstanter Spektraldichtematrix des Systemrauschens (Variante 1 in Tafel 3.2).
- Methode `'lin'`*: Numerische Integration der Variationsgleichungen (3.44), lineare Näherung (3.51) für $\boldsymbol{Q}_k$ mit konstanten Werten für $\boldsymbol{Q}$ und zeitdiskrete Prädiktion gemäß Gl. (3.37) der zweiten Variante in Tafel 3.2.
- Methode `'const'`*: Numerische Integration der Variationsgleichungen (3.44), konstante Diagonalmatrix $\boldsymbol{Q}_k \; \forall t \in (t_k, t_{k+1}]$ und zeitdiskrete Prädiktion gemäß Gl. (3.37) der zweiten Variante in Tafel 3.2.

Es wird also stets von einem numerisch zu lösenden DGL-System ausgegangen, entweder zur Bestimmung der Transitionsmatrix $\boldsymbol{\Phi}(t, t_k)$ (Variationsgleichungen) oder der Kovarianzmatrix $\bar{\boldsymbol{P}}(t)$ direkt (Riccati-DGL), wobei der Rechenaufwand vergleichbar ist, da die Anzahl der DGLen in beiden Fällen gleich ist. Eine Ausnahme bildet das einfache, ungestörte Zweikörperproblem, weil hierbei geschlossene, analytische Formeln für die Bestimmung zukünftiger Satellitenpositionen und -geschwindigkeiten und der Transitionsmatrix existieren (s. nächster Abschnitt). Bei den Methoden `'lin'` und `'const'` wurde auf die näherungsweise Berechnung der Transitionsmatrix mithilfe der Matrixexponentialfunktion entsprechend Gl. (3.47) verzichtet.

Die Methode `'lin'` stellt eine Approximation der Methode `'ric'` dar. Bei beiden Methoden steigt das Systemrauschen mit der Integrationszeit an (bei `'lin'` linear). Anhand Gleichung (3.51) ist dies eben deutlicher zu erkennen als anhand der Riccati-DGL. Bei der Implementierung wird später dafür Sorge getragen, dass die innerhalb eines Integrationsintervalls zuletzt prädizierte Kovarianzmatrix bei allen Methoden ungefähr denselben Anteil an Systemrauschen aufweist, sodass die drei Methoden vergleichbare Ergebnisse liefern. Der wesentliche Unterschied zwischen den

∗ Diese Bezeichnungen für die Methoden werden hier eingeführt, weil man damit auf sie Bezug nimmt bei der Implementierung und in diversen Quelltexten der eigenen Toolbox.

Methoden besteht somit darin, ob innerhalb des Integrationsintervalls von einem ansteigenden Systemrauschen auszugehen ist (`'ric'`/`'lin'`) oder nicht (`'const'`), was sich entsprechend auf die Kovarianzen der prädizierten Zustandsvektoren auswirkt.

Entscheidend ist bei allen Methoden stets, dass die Zahlenwerte der Diagonalenelemente der $\boldsymbol{Q}$-Matrix angemessen gewählt werden, denn dies hat beim Korrekturschritt einen nicht unerheblichen Einfluss auf die Gewichtung zwischen der prädizierten und der tatsächlichen Beobachtung und damit auf die Filterstabilität.

3.2.4.2 Filterstabilität

Der Stabilität des Kalman-Filters ist die Divergenz entgegengesetzt – eine Tendenz, bei der sich die geschätzten Systemzustände immer weiter vom wahren bzw. vom gemessenen Zustand entfernen. Dabei stellt sich zunächst aber die Frage, weshalb die prädizierte von der tatsächlichen Messung überhaupt abweicht. Freilich sind die Beobachtungen an sich immer mit Unsicherheiten verbunden. Abgesehen davon kann zum einen das analytisch zu bestimmende Messmodell (Funktion $\boldsymbol{h}$ und Messmatrix $\boldsymbol{H}$) fehlerhaft sein. Andrerseits kann es auch vorkommen, dass das verwendete Kräftemodell das Systemverhalten nicht hinreichend genau beschreibt. Obendrein könnten falsch gewählte Anfangsbedingungen oder ein unzureichendes numerisches Integrationsverfahren die Abweichung begünstigen.

Entscheidend für das Aufkommen der Divergenz, d.h. den Anstieg dieser Abweichung, ist insbesondere, ob die Einträge der entsprechenden Kovarianzmatrix genügend groß sind und damit sozusagen auf den jeweiligen Fehlereinfluss hindeuten. Im Zusammenhang mit den Kovarianzmatrizen können typischerweise zweierlei Schwierigkeiten auftreten:

1. Aus Gleichung (3.36) und (3.37) für die Prädiktion der Schätzfehlerkovarianzmatrix ist klar geworden, dass die prädizierte Kovarianzmatrix $\bar{\boldsymbol{P}}$ umso größer wird, je größer die Matrixelemente des Systemrauschens $\boldsymbol{Q}$ gewählt werden. Falls man letztere jedoch zu klein annimmt, werden die Diagonalelemente der korrigierten Kovarianzmatrix $\widehat{\boldsymbol{P}}$ aufgrund Gl. (3.18) nach einigen Filterdurchläufen deutlich kleiner. Lax ausgedrückt ‚meint' das Filter dann schlimmstenfalls, es ‚kenne' den Zustand bereits sehr genau und ignoriert neu hinzukommende Messungen. Sämtliche bei der Prädiktion gemachten Fehler (Approximations- und Rundungsfehler, Vereinfachungen beim Kräftemodell) werden dann ungehindert mitintegriert, ohne dass die eigentlich nötige Korrektur durch Messungen stattfindet. Somit entfernt sich der prädizierte Zustandsvektor immer weiter von seinem tatsächlichen Wert, d.h. das Filter divergiert.
2. Das andere Extrem tritt auf, wenn die prädizierten Kovarianzen ($\bar{\boldsymbol{P}}$-Matrix) viel größer sind als die des gemessenen Zustands ($\boldsymbol{R}$-Matrix). Dann wird hauptsächlich der Messung vertraut, ungeachtet des auftretenden Messrauschens. Dies bewirkt zwar keine unmittelbare Divergenz, der korrigierte Systemzustand ist aber kaum genauer als der beobachtete Wert. In diesem Fall wird also gewissermaßen der prädizierte Zustand bei der Korrektur überkompensiert.

Die Zahlenwerte der Diagonalenelemente in der $\boldsymbol{Q}_k$- bzw. $\boldsymbol{Q}$-Matrix werden normalerweise durch simples Ausprobieren so lange angepasst, bis das Gleichgewicht zwischen System- und Messrauschen gefunden ist („trial and error"-Prinzip). Auf ähnliche Weise bestimmt man auch hier geeignete Zahlenwerte bei der Verarbeitung simulierter bzw. realer Daten. Wahrscheinlich wäre dieser Vorgang bei der Technik nach Wright et al. (2008) zur Bestimmung der Kovarianzmatrix des Systemrauschens nicht notwendig.

3.2.5 Kräftemodelle zur Prädiktion der Satellitenbahn

Im Folgenden betrachtet man die für den Prädiktionsschritt erforderliche Orbitintegration und insbesondere die dabei verwendeten Kräftemodelle, die in dieser Arbeit umgesetzt wurden.

Bei der Prädiktion ist der erste bzw. der zuletzt korrigierte Anfangszustand $\hat{\boldsymbol{s}}_k$ und dessen Kovarianzmatrix $\widehat{\boldsymbol{P}}_k$ zum Zeitpunkt t_k gegeben. Gesucht sind die prädizierten Zustandsvektoren $\bar{\boldsymbol{s}}(t)$ und die zugehörigen Kovarianzmatrizen $\bar{\boldsymbol{P}}(t)$, und zwar zunächst nur für den Bahnabschnitt bis zum darauffolgenden Beobachtungszeitpunkt t_{k+1}. Die gesuchten Größen sind im Prädiktionsschritt also stets nur für einen kurzen Bahnbogen der ‚Länge' $t_{k+1} - t_k$ zu ermitteln.

Entsprechend der obigen Festlegungen ist dazu ein DGL-System (erster Ordnung) zu lösen. Dessen erster Teil besteht aus der nicht-linearen Systemgleichung (3.22). Den zweiten Teil des Gleichungssystems bilden entweder die DGLen zur Bestimmung der Kovarianzmatrizen $\bar{\boldsymbol{P}}(t)$ (im Fall `'ric'`) oder die DGLen zur Bestimmung der Transitionsmatrizen $\boldsymbol{\Phi}(t, t_k)$ (im Fall `'lin'` und `'const'`). Je nach Kräftemodell unterscheidet sich die darin vorkommende Kräftefunktion $\boldsymbol{f}_\mathrm{K}$ und die Jacobi-Matrix $\boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})$. Zu jedem Kräftemodell wurde ein Orbitintegrator umgesetzt, der die zukünftigen Bahnvektoren und die Transitionsmatrizen oder wahlweise die Kovarianzmatrizen $\bar{\boldsymbol{P}}(t)$ durch numerische Integration des entsprechenden DGL-Systems berechnet. Eine Ausnahme bildet der Spezialfall des Zweikörperproblems, weil dabei anstatt der numerischen Lösungen zum Teil analytische Lösungen möglich sind, die später noch aufgezeigt werden.

Numerische Verfahren zur Satellitenbahnbestimmung

Die numerische Lösung basiert hier auf der sogenannten *Methode nach Cowell*[1], wonach die gesamte auf den Satelliten wirkende Beschleunigung schrittweise integriert wird, inklusive aller berücksichtigten Störeinflüsse (Seeber, 1989, S. 119). Prinzipiell wäre auch das *Verfahren nach Encke*[2], ein weiteres klassisches Bahnbestimmungsverfahren, anwendbar, wobei man zumeist eine Keplerbahn als Referenz wählt und nur die Differenzbeschleunigung zur gestörten Bahn integriert. Da die Abweichungen zwischen der Referenzbahn und der gestörten Bahn in der Regel klein sind, können größere Schrittweiten bei der Integration der Differenzbeschleunigungen ausreichen. Übersteigen nach einer gewissen Integrationszeit die Abweichungen zwischen den Bahnen vorab festgelegte Toleranzwerte, wird am Bahnpunkt der gestörten Bahn eine neue oskulierende Ellipse als Referenzbahn gewählt (sog. Rektifizierung). Dies erinnert stark an den bereits kurz umrissenen Berechnungsablauf des linearisierten Kalman-Filters. Das LKF unterscheidet sich jedoch vor allem dadurch, dass die Abweichungen mithilfe von Messungen korrigiert werden. Aufgrund der größeren Integrationsschritte bei dem Verfahren nach Encke erhofft man sich, dass die gesamte Rechenzeit kürzer ausfällt im Vergleich zur Methode nach Cowell. Zudem ist beim Verfahren nach Encke mit geringeren Rundungsfehlern bei der Integration zu rechnen, da die integrierten Werte als Differenzen betragsmäßig kleiner sind und so mehr Nachkommastellen bei der Berechnung zur Verfügung stehen. Andererseits wirken sich aber wegen der größeren Schrittweite Abbruchfehler stärker aus[3].
Angesichts der heutzutage verfügbaren schnellen und leistungsfähigen Computersysteme mit tendenziell ansteigenden Wortbreiten (derzeit 32 od. 64 Bit) sind Rundungsfehler meist weniger problematisch zu sehen, weshalb heute oftmals – zumindest bei Anwendungen mit hohen Genauigkeitsansprüchen – die direkte numerische Integration entsprechend der Methode nach Cowell bevorzugt wird. Hier wurde sie auch deshalb gewählt, weil die DGLen dann eine einfachere Struktur haben. Zudem stellt der Rechenzeitbedarf kein ausschlaggebendes Kriterium in einer Postprocessing-Anwendung wie dieser dar.

1 Entwickelt um 1910 von Philip Herbert Cowell (1870-1949), brit. Astronom.
2 Entwickelt um 1857 von Johann Franz Encke (1791-1865), dt. Astronom.
3 Abbruchfehler kommen durch den Abbruch der bei der Integration verwendeten Reihenentwicklungen zustande.

Generelle Struktur der dynamischen Größen

Zunächst werden nun die für den Prädiktionsschritt festzulegenden dynamischen Größen (Kraftfunktion $\boldsymbol{f}_{\mathrm{K}}$ und Jacobi-Matrix $\boldsymbol{F}$) übergeordnet bzw. allgemein für alle Kräftemodelle dieser Arbeit betrachtet.

Gemäß Gl. (3.22) hat die nicht-lineare Systemfunktion die Form

$$\boldsymbol{f}(t,\boldsymbol{s}(t),\boldsymbol{p}) = \begin{pmatrix} \dot{\boldsymbol{X}}(t) \\ \boldsymbol{f}_{\mathrm{K}}(t,\boldsymbol{X}(t),\boldsymbol{p}) \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{X}}(t) \\ \nabla_{\mathrm{I}} V \end{pmatrix}, \tag{3.52}$$

worin $\boldsymbol{f}_{\mathrm{K}}(t,\boldsymbol{X}(t),\boldsymbol{p})$ die je nach Kräftemodell spezifische Kräftefunktion bezeichnet. Diese Kräftefunktion ist, wie schon erwähnt, bei allen in dieser Arbeit verwendeten Kräftemodellen nicht von der Geschwindigkeit des Satelliten abhängig. Sie entspricht dem Gradienten des im jeweiligen Kräftemodell angenommenen Gravitationspotentials V, der im raumfesten, geozentrischen Bezugssystem gebildet wird (Index „I“). Der Ausdruck $\nabla_{\mathrm{I}} V$ ist im Fall eines a priori Gravitationsfeldmodells von der Zeit, der Satellitenposition und weiteren Modellparametern $\boldsymbol{p}$ abhängig (s. nächster Abschnitt) und bei der Orbitintegration am aktuellen Systemzustand $\bar{\boldsymbol{s}}(t)$ zu berechnen.

Die Jacobi-Matrix $\boldsymbol{F}(t,\bar{\boldsymbol{s}}(t),\boldsymbol{p})$ ist nach Gleichung (3.30) gegeben durch

$$\boldsymbol{F}(t,\bar{\boldsymbol{s}}(t),\boldsymbol{p}) = \left.\frac{\partial \boldsymbol{f}(t,\boldsymbol{s}(t),\boldsymbol{p})}{\partial \boldsymbol{s}(t)}\right|_{\bar{\boldsymbol{s}}(t)}$$

und weist dementsprechend folgende Struktur auf:

$$\boldsymbol{F}(t,\bar{\boldsymbol{s}}(t),\boldsymbol{p}) = \left.\begin{pmatrix}
\frac{\partial \dot{X}}{\partial X} & \frac{\partial \dot{X}}{\partial Y} & \frac{\partial \dot{X}}{\partial Z} & \frac{\partial \dot{X}}{\partial \dot{X}} & \frac{\partial \dot{X}}{\partial \dot{Y}} & \frac{\partial \dot{X}}{\partial \dot{Z}} \\
\frac{\partial \dot{Y}}{\partial X} & \frac{\partial \dot{Y}}{\partial Y} & \frac{\partial \dot{Y}}{\partial Z} & \frac{\partial \dot{Y}}{\partial \dot{X}} & \frac{\partial \dot{Y}}{\partial \dot{Y}} & \frac{\partial \dot{Y}}{\partial \dot{Z}} \\
\frac{\partial \dot{Z}}{\partial X} & \frac{\partial \dot{Z}}{\partial Y} & \frac{\partial \dot{Z}}{\partial Z} & \frac{\partial \dot{Z}}{\partial \dot{X}} & \frac{\partial \dot{Z}}{\partial \dot{Y}} & \frac{\partial \dot{Z}}{\partial \dot{Z}} \\
\frac{\partial f_{\mathrm{K},X}}{\partial X} & \frac{\partial f_{\mathrm{K},X}}{\partial Y} & \frac{\partial f_{\mathrm{K},X}}{\partial Z} & \frac{\partial f_{\mathrm{K},X}}{\partial \dot{X}} & \frac{\partial f_{\mathrm{K},X}}{\partial \dot{Y}} & \frac{\partial f_{\mathrm{K},X}}{\partial \dot{Z}} \\
\frac{\partial f_{\mathrm{K},Y}}{\partial X} & \frac{\partial f_{\mathrm{K},Y}}{\partial Y} & \frac{\partial f_{\mathrm{K},Y}}{\partial Z} & \frac{\partial f_{\mathrm{K},Y}}{\partial \dot{X}} & \frac{\partial f_{\mathrm{K},Y}}{\partial \dot{Y}} & \frac{\partial f_{\mathrm{K},Y}}{\partial \dot{Z}} \\
\frac{\partial f_{\mathrm{K},Z}}{\partial X} & \frac{\partial f_{\mathrm{K},Z}}{\partial Y} & \frac{\partial f_{\mathrm{K},Z}}{\partial Z} & \frac{\partial f_{\mathrm{K},Z}}{\partial \dot{X}} & \frac{\partial f_{\mathrm{K},Z}}{\partial \dot{Y}} & \frac{\partial f_{\mathrm{K},Z}}{\partial \dot{Z}}
\end{pmatrix}\right|_{\bar{\boldsymbol{s}}(t)} \tag{3.53}$$

Die partiellen Ableitungen sind zeitabhängig und ebenfalls stets am momentan prädizierten Systemzustand $\bar{\boldsymbol{s}}(t)$ auszuwerten. Da die Kräftefunktion unabhängig von der Satellitengeschwindigkeit ist, gilt

$$\boldsymbol{F}(t,\bar{\boldsymbol{s}}(t),\boldsymbol{p}) = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\left.\frac{\partial f_{\mathrm{K},X}(t,\boldsymbol{X}(t))}{\partial X}\right|_{\bar{\boldsymbol{s}}(t)} & \left.\frac{\partial f_{\mathrm{K},X}(t,\boldsymbol{X}(t))}{\partial Y}\right|_{\bar{\boldsymbol{s}}(t)} & \left.\frac{\partial f_{\mathrm{K},X}(t,\boldsymbol{X}(t))}{\partial Z}\right|_{\bar{\boldsymbol{s}}(t)} & 0 & 0 & 0 \\
\left.\frac{\partial f_{\mathrm{K},Y}(t,\boldsymbol{X}(t))}{\partial X}\right|_{\bar{\boldsymbol{s}}(t)} & \left.\frac{\partial f_{\mathrm{K},Y}(t,\boldsymbol{X}(t))}{\partial Y}\right|_{\bar{\boldsymbol{s}}(t)} & \left.\frac{\partial f_{\mathrm{K},Y}(t,\boldsymbol{X}(t))}{\partial Z}\right|_{\bar{\boldsymbol{s}}(t)} & 0 & 0 & 0 \\
\left.\frac{\partial f_{\mathrm{K},Z}(t,\boldsymbol{X}(t))}{\partial X}\right|_{\bar{\boldsymbol{s}}(t)} & \left.\frac{\partial f_{\mathrm{K},Z}(t,\boldsymbol{X}(t))}{\partial Y}\right|_{\bar{\boldsymbol{s}}(t)} & \left.\frac{\partial f_{\mathrm{K},Z}(t,\boldsymbol{X}(t))}{\partial Z}\right|_{\bar{\boldsymbol{s}}(t)} & 0 & 0 & 0
\end{pmatrix}. \tag{3.54}$$

Es zeigt sich, dass die linke untere (3×3)-Submatrix der sogenannte Schweregradiententensor des im jeweiligen Kräftemodell angenommenen Gravitationspotentials V ist, formuliert im raumfesten,

geozentrischen Bezugssystem. Denn die Kräftefunktion $\boldsymbol{f}_\mathrm{K}$ ist nach Gleichung (3.52) mit dem Gradienten des Gravitationspotentials identisch. Es ist leicht ersichtlich, dass die gesuchte Submatrix der Jacobi-Matrix durch zweimaliges Anwenden des Gradientenoperators auf das Gravitationspotential folgt. Dies ist gerade die Definition des Schweregradiententensors (auch Marussi- oder Eötvös-Tensor) in geozentrischen, raumfesten Koordinaten:

$$\boldsymbol{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} = \nabla^\top \otimes \nabla_\mathrm{I} V = \begin{pmatrix} \frac{\partial^2 V}{\partial X^2} & \frac{\partial^2 V}{\partial X \partial Y} & \frac{\partial^2 V}{\partial X \partial Z} \\ \frac{\partial^2 V}{\partial Y \partial X} & \frac{\partial^2 V}{\partial Y^2} & \frac{\partial^2 V}{\partial Y \partial Z} \\ \frac{\partial^2 V}{\partial Z \partial X} & \frac{\partial^2 V}{\partial Z \partial Y} & \frac{\partial^2 V}{\partial Z^2} \end{pmatrix} \tag{3.55}$$

Abgekürzt gilt somit allgemein für die hier betrachteten Kräftemodelle

$$\boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) = \begin{pmatrix} \mathbf{0}_{(3\times3)} & \mathbf{I}_{(3\times3)} \\ \boldsymbol{M}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) & \mathbf{0}_{(3\times3)} \end{pmatrix}. \tag{3.56}$$

Da das Gravitationsfeld V in allen nachfolgenden Kräftemodellen wirbelfrei und damit konservativ ist, d.h.

$$\mathrm{rot}\,(\nabla V) = 0,$$

sind die Elemente des Marussi-Tensors symmetrisch zur Hauptdiagonalen. Somit sind folgende Matrixelemente bzw. zweite Ableitungen äquivalent:

$$\begin{aligned} M_{21} &= \frac{\partial^2 V}{\partial y \partial x} = \frac{\partial^2 V}{\partial x \partial y} = M_{12} \\ M_{31} &= \frac{\partial^2 V}{\partial z \partial x} = \frac{\partial^2 V}{\partial x \partial z} = M_{13} \\ M_{32} &= \frac{\partial^2 V}{\partial z \partial y} = \frac{\partial^2 V}{\partial y \partial z} = M_{23}. \end{aligned}$$

Das Gravitationsfeld V ist hier außerdem stets quellenfrei, weshalb es (im betrachteten Außenraum) der bekannten Laplace-Bedingung

$$\Delta V = \mathrm{Sp}(\boldsymbol{M}) = \frac{\partial^2 V}{\partial X^2} + \frac{\partial^2 V}{\partial Y^2} + \frac{\partial^2 V}{\partial Z^2} = 0 \tag{3.57}$$

genügt, sodass insgesamt nur fünf der neun Matrixelemente des Marussi-Tensors voneinander unabhängig sind.

Im Folgenden ist für jedes Kräftemodell also nur noch der Gradient des Gravitationspotentials (Kräftefunktion) sowie der Marussi-Tensor (beides bezüglich des raumfesten Systems) formelmäßig zu spezifizieren.

3.2.5.1 Kräftemodell für die Satellitenbewegung im Erdgravitationsfeld

Bei diesem Kräftemodell geht man davon aus, dass die Beschleunigung, die der Satellit erfährt, nur durch das Erdschwerefeld zustande kommt. Dementsprechend wird in diesem Abschnitt der Formelapparat zusammengestellt, der nötig ist, um die gesuchten dynamischen Größen auf Grundlage eines a priori Erdgravitationsfeldmodells bis zu einem bestimmten Maximalgrad zu bestimmen. Wie dies in Matlab (in der Funktion `orbint_field`) umgesetzt wurde, ist später (in Abschnitt 4.4) dokumentiert.

Das statische Schwerepotential des Erdschwerefelds setzt sich im erdfesten System bekanntlich aus dem Gravitations- und dem Zentrifugalpotential zusammen:

$$W = V + V_{\mathrm{Z}}.$$

Da die zu integrierende Bewegungsgleichung des Satelliten hier im *raumfesten* System formuliert wird, ist lediglich das stationäre Gravitationsfeld mit dem Gravitationspotential V von Interesse. Außerdem können, wie bereits erläutert, bei der Beschreibung der Satellitenbewegung im raumfesten System sämtliche Scheinkräfte, wie etwa der durch die Satellitenbewegung scheinbar für einen erdfesten Beobachter festzustellende Coriolis-Effekt, entfallen. Das Gravitationspotential V lässt sich allerdings nicht bezüglich des raumfesten, sondern bezüglich eines lokalen Horizontsystems vergleichsweise einfach beschreiben, da das Gravitationsfeld unmittelbar an den (rotierenden) Erdkörper gebunden ist. Gebräuchlich ist folgende, aus der physikalischen Geodäsie bekannte Darstellung des Gravitationspotentials der Erde in Form einer Reihenentwicklung nach Kugelfunktionen (vgl. Hofmann-Wellenhof und Moritz, 2006, Abschn. 1.5, S. 11 ff.):

$$V(r,\varphi,\lambda,\boldsymbol{p}) = \frac{GM}{R_{\mathrm{E}}} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l+1} \cdot \bar{P}_{l,m}(\sin(\varphi)) \cdot \left[\bar{C}_{l,m} \cdot \cos(m\lambda) + \bar{S}_{l,m} \cdot \sin(m\lambda)\right] \tag{3.58}$$

Darin bedeuten:

$r,\ \varphi,\ \lambda$	Kugelkoordinaten der Satellitenposition im erdfesten, geozentr. Bezugssystem
$\bar{P}_{l,m}(\sin(\varphi))$	vollständig normierte Legendre-Funktion (erster Art) von Grad l und Ordnung m
	Parameter $\boldsymbol{p}$ des Gravitationsfeldmodells:
GM	Geozentrische Gravitationskonstante, hier $GM = 3986004.415 \cdot 10^{14}\ \mathrm{m^3/s^2}$
R_{E}	Äquatorradius, hier $R_{\mathrm{E}} = a_{\mathrm{E}} = 6378136.3\,\mathrm{m}$
$\bar{C}_{l,m},\ \bar{S}_{l,m}$	vollständig normierte Kugelfunktionskoeffizienten (von Grad l und Ordnung m)
$l_{\max}$	Maximaler Entwicklungsgrad.

Die Produkte

$$\bar{P}_{l,m}(\sin(\varphi)) \begin{Bmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{Bmatrix}$$

stellen dabei die Kugelflächenfunktionen dar. Die Entwicklung (3.58) gilt nur im Außenraum der Erde ($r \geq R_{\mathrm{E}}$), welcher durch die sog. Brillouin-Kugel mit dem Radius R_{E} vom Innenraum abgegrenzt ist. Die Brillouin-Kugel umschließt die Erde und damit die felderzeugenden Massen vollständig (Hofmann-Wellenhof und Moritz, 2006, S. 60). Daher wählt man für R_{E} meist die große Halbachse a_{E} eines zugrundegelegten mittleren Erdellipsoids. Die einheitenlosen, harmonischen Koeffizienten $\bar{C}_{l,m},\ \bar{S}_{l,m}$ können in obiger Außenraumentwicklung als Maßzahlen für die innere Massenverteilung des Zentralkörpers, d.h. der Erde, verstanden werden (Seeber, 1989, S. 91). Zusammen mit der Halbachse a_{E}, der Gravitationskonstante GM und des Maximalgrads $l_{\max}$ gehören sie zu den Parametern $\boldsymbol{p}$ des statischen Gravitationsfeldmodells. Je nach Modell können diese Parameter unterschiedliche Zahlenwerte haben. GM und a_{E} sind jedoch bei allen hier genutzten Modellen identisch.

Man verwendet hier die Kugelfunktionsentwicklung des Gravitationspotentials mit den *vollständig normierten* Legendre-Funktionen und ebensolchen Potentialkoeffizienten. Dies hat u.a. den Vorteil, dass die Orthonormalität der Kugelflächenfunktionen sichergestellt ist (vgl. Hofmann-Wellenhof und Moritz, 2006, S. 23). Zwischen den vollständig normierten und den unnormierten Größen bestehen folgende Zusammenhänge (s. Hofmann-Wellenhof und Moritz, 2006, Gl. 1-91

und Gl. 1-98):

$$\bar{P}_{l,m}(\sin(\varphi)) = \eta_{l,m} P_{l,m}(\sin(\varphi)) \tag{3.59}$$

$$C_{l,m,j} = \eta_{l,m} \bar{C}_{l,m,j} \tag{3.60}$$

$$\eta_{l,m} = \begin{cases} \sqrt{2(2 \cdot l + 1)\frac{(l-m)!}{(l+m)!}}, & m > 0 \\ \sqrt{2 \cdot l + 1}, & m = 0 \end{cases} \quad \text{(Normierungsfaktor)} \tag{3.61}$$

Es liegt auf der Hand, dass bei einer Orbitverdichtung im Gegensatz zur Gravitationsfeldbestimmung die sphärisch-harmonischen Kugelfunktionskoeffizienten nicht als Unbekannte aufzufassen sind. Vielmehr müssen die Potentialkoeffizienten durch ein vorhandenes Gravitationsfeldmodell der Erde als a priori Information geliefert werden, sodass sich aus Gleichung (3.58) der Wert des Gravitationspotentials an jedem Ort des Außenraums ergibt.

Die Koeffizienten des zugrundegelegten Schwerefeldmodells stehen nur bis zu einem bestimmten Entwicklungsgrad zur Verfügung. Ratsam ist jedoch, die Reihenentwicklung bereits vorher bei einem selbst gewählten Maximalgrad $l_{\max}$ abzuberechen, da einerseits die Koeffizienten des Schwerefeldmodells an sich Unsicherheiten aufweisen, andererseits der Rechenaufwand bei der numerischen Integration umso größer ist, je höher der Entwicklungsgrad gewählt wird.

In Tabelle 3.1 sind die globalen Schwerefeldmodelle aufgeführt, die für die Orbitintegration mit der eigenen Toolbox verwendet werden können. Diese stehen als `mat`-Dateien im Ordner `gravfields` zur Verfügung, allerdings enthalten sie jeweils nur die vollständig normierten Potentialkoeffizienten bis maximal Grad/Ordnung 160[1].

Tab. 3.1: Globale Gravitationsfeldmodelle

Modell	Max. Grad (Auflösung)	Bezugsquelle/ Organisation	Aktualität	Literatur
EGM08	2159 (2190)	NGA[2]	4/2008	Pavlis et al. (2008)
EGM96	360	NASA[3], NIMA, OSU	1996	Lemoine et al. (1998)
GGM02s	160	UTCSR[4]	12/2003	Tapley et al. (2005)
JGM-3	70	UTCSR[5], NASA, HSTX	1994	Tapley et al. (1996)

Zu den Modellen JGM-3 und GGM02s ist anzumerken, dass sie ursprünglich im „Null-Gezeiten"-(engl.: „zero-tide"-) System gegeben sind, d.h. der permanente Anteil der Gezeiten der festen Erde ist im Koeffizienten $\bar{C}_{2,0}$ enthalten (s. Petit und Luzum, 2010, Abschn. 6.2.2., S. 88). Demgegenüber sind EGM96 und EGM08 ursprünglich im gezeitenfreien („tide-free"-) System gegeben, wobei der Potentialkoeffizient durch ein hypothesenbehaftetes Modell um den permanenten Anteil der Erdgezeiten reduziert wurde, sodass der Koeffizient keinerlei Gezeiteneinflüsse mehr repräsentiert. Als dritte Variante ist das mittlere („mean-tide"-) System zu nennen, das neben dem permanenten Anteil der Erdgezeiten auch den mittleren Beitrag der direkten Gezeiten (durch eine eventuell unrealistische Mittelwertbildung) einbezieht (s.a. Torge, 2001, S. 77). In den IERS-Konventionen wird nahegelegt, bei Größen, die mit dem Gravitationspotential verknüpft sind, die „zero-tide"-Koeffizienten zu verwenden (Petit und Luzum, 2010, S. 15/16). Dementsprechend werden hier auch bei den Modellen EGM96 und EGM08 jeweils die „zero-tide"-Werte für $\bar{C}_{2,0}$ genutzt.

Neben den zeitunabhängigen Einflüssen müssten eigentlich auch Korrekturen aufgrund der Zeitvariabilität des Schwerefeldes berücksichtigt werden, z.B. die säkularen Änderungsraten bestimmter

1 Noch höhere Grade wären – abgesehen von der eher unhandlichen Dateigröße – kaum rentabel, da andere Einflüsse den Orbit weitaus mehr stören.
2 http://earth-info.nima.mil/GandG/wgs84/gravitymod/egm2008/first_release.html
3 http://cddis.gsfc.nasa.gov/926/egm96/egm96.html
4 http://www.csr.utexas.edu/grace/gravity/
5 ftp://ftp.csr.utexas.edu/pub/grav/JGM3.GEO

Potentialkoeffizienten (s. PETIT und LUZUM, 2010, Gl. 6.4, S. 80). Wegen der kurzen Prädiktionszeiten von 5 bzw. 60 Sekunden können solche Korrekturen hier jedoch vernachlässigt werden.

Für die Orbitintegration benötigt man nun nicht den Wert des Gravitationspotentials selbst, sondern dessen erste und zweite partielle Ableitungen in kartesischen Koordinaten, zunächst bezüglich des erdfesten Systems. Hierfür gibt es prinzipiell zwei verschiedene Ansätze (SCHMIDT, 1999, S. 21):

(i) Partielles Differenzieren von Gleichung (3.58) nach Kugelkoordinaten mit Nachdifferenzieren nach kartesischen Koordinaten entsprechend der Kettenregel. Laut REED (1973, S. 63), der auch die betreffenden allgemeinen Formeln für die ersten und zweiten Ableitungen von V angibt, stammt dieser Ansatz von GULICK (1970).

(ii) Gradienten- bzw. Tensorbildung in Kugelkoordinaten bezüglich eines lokalen Satelliten-Koordinatensystems und anschließende Transformation durch geeignete Drehmatrizen in das erdfeste, geozentrische System.

REED (1973, S. 65) zufolge liefern beide Techniken prinzipiell identische Ergebnisse. Hier wurde die Tensormethode nach Punkt (ii) gewählt. Dabei wird der Gradient des Gravitationspotentials als Tensor erster Stufe aufgefasst, der sich gemäß den Vorschriften der Tensorrechnung herleiten lässt. Das lokale Horizontsystem, dessen Ursprung mit der aktuellen Satellitenposition festgelegt wird, stellt bezüglich der Parametrisierung des Gravitationspotentials in krummlinigen Kugelkoordinaten eine ortsabhängige Basis dar (SCHEINERT, 1996, S. 26). Die Basisvektoren dieses Systems sind durch die normierten kovarianten Tangentenvektoren

$$\boldsymbol{e}_r = \frac{\partial \boldsymbol{x}}{\partial r}\left\|\frac{\partial \boldsymbol{x}}{\partial r}\right\|^{-1} \qquad \boldsymbol{e}_\varphi = \frac{\partial \boldsymbol{x}}{\partial \varphi}\left\|\frac{\partial \boldsymbol{x}}{\partial \varphi}\right\|^{-1} \qquad \boldsymbol{e}_\lambda = \frac{\partial \boldsymbol{x}}{\partial \lambda}\left\|\frac{\partial \boldsymbol{x}}{\partial \lambda}\right\|^{-1} \tag{3.62}$$

definiert (REUBELT, 2009, S. 43). Wählt man folgende (linkshändige) Achsausrichtung (s. Abb. 3.4)

- x_L-Achse: Normalenvektor $\boldsymbol{e}_r$ nach außen in Richtung r
- y_L-Achse: Tangentenvektor $\boldsymbol{e}_\varphi$ nach Norden in Richtung φ
- z_L-Achse: Tangentenvektor $\boldsymbol{e}_\lambda$ nach Osten in Richtung λ,

so ergeben sich die Tangentenvektoren durch die Ableitungen der bekannten Gleichungen

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos(\varphi)\cos(\lambda) \\ r\cos(\varphi)\sin(\lambda) \\ r\sin(\varphi) \end{pmatrix} \tag{3.63}$$

und man erhält damit die Transformationsbeziehungen

$$\begin{pmatrix} x_\mathrm{L} \\ y_\mathrm{L} \\ z_\mathrm{L} \end{pmatrix} = \begin{pmatrix} \cos(\varphi)\cos(\lambda) & \cos(\varphi)\sin(\lambda) & \sin(\varphi) \\ -\sin(\varphi)\cos(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\varphi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{R}_\mathrm{L}^\mathrm{E}(\varphi,\lambda)\cdot\boldsymbol{x} \tag{3.64}$$

zwischen den kartesischen erdfesten Koordinaten ($\boldsymbol{x}$) und den kartesischen Koordinaten des lokalen Horizontsystems ($\boldsymbol{x}_\mathrm{L}$). Die Transformation kann auch anhand einzelner Rotationen wie folgt dargestellt werden:

$$\boldsymbol{x}_\mathrm{L} = \boldsymbol{P}_3\boldsymbol{R}_1(\pi/2)\boldsymbol{R}_2(-\varphi)\boldsymbol{R}_3(\lambda)\boldsymbol{x}. \tag{3.65}$$

Dabei bezeichnet $\boldsymbol{P}_3 = \mathrm{diag}(1,1,-1)$ die Matrix für eine Spiegelung der dritten Achse und $\boldsymbol{R}_1$ bis $\boldsymbol{R}_3$ sind die bekannten Rotationsmatrizen für ein rechtshändiges dreidimensionales Koordinatensystem.

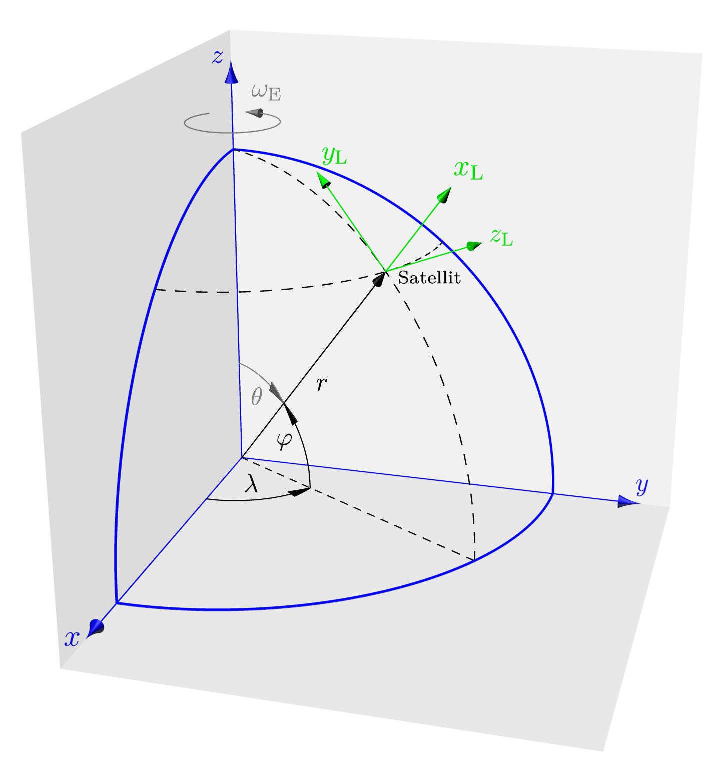


Abb. 3.4: Festlegung des lokalen Horizontsystems im Satellitenaufpunkt

Der gesuchte Gradient des Gravitationspotentials lautet in dem so gewählten Horizontsystem (vgl. z.B. Reed, 1973, S. 28, Gl. 3.17; Scheinert, 1996, S. 27; Schäfer, 2000, S. 34):

$$\nabla_{\mathrm{L}} V(r, \varphi, \lambda, \boldsymbol{p}) = \left(\frac{\partial V}{\partial r}, \; \frac{1}{r}\frac{\partial V}{\partial \varphi}, \; \frac{1}{r\cos(\varphi)}\frac{\partial V}{\partial \lambda} \right)^{\top} \tag{3.66}$$

Die darin enthaltenen partiellen Ableitungen von V nach den erdfesten, sphärischen Koordinaten ergeben sich mit der Kugelfunktionsentwicklung (3.58) zu (s. z.B. Schäfer, 2000, S. 34; Götzelmann, 2003, S. 39; Meyer, 2006, S. 101):

$$\begin{aligned}
\frac{\partial V}{\partial r} &= \frac{GM}{R_{\mathrm{E}}} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l+1} \left(-\frac{(l+1)}{r}\right) \bar{P}_{l,m}(\sin(\varphi)) \left[\bar{C}_{l,m}\cos(m\lambda) + \bar{S}_{l,m}\sin(m\lambda)\right] \\
&= -\frac{GM}{r^2} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l} (l+1)\bar{P}_{l,m}(\sin(\varphi)) \left[\bar{C}_{l,m}\cos(m\lambda) + \bar{S}_{l,m}\sin(m\lambda)\right]
\end{aligned} \tag{3.67}$$

$$\begin{aligned}
\frac{\partial V}{\partial \varphi} &= \frac{GM}{R_{\mathrm{E}}} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l+1} \frac{\partial \bar{P}_{l,m}(\sin(\varphi))}{\partial \varphi} \left[\bar{C}_{l,m}\cos(m\lambda) + \bar{S}_{l,m}\sin(m\lambda)\right] \\
&= \frac{GM}{r} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l} \frac{\partial \bar{P}_{l,m}(\sin(\varphi))}{\partial \varphi} \left[\bar{C}_{l,m}\cos(m\lambda) + \bar{S}_{l,m}\sin(m\lambda)\right]
\end{aligned} \tag{3.68}$$

$$\begin{aligned}
\frac{\partial V}{\partial \lambda} &= \frac{GM}{R_{\mathrm{E}}} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l+1} \bar{P}_{l,m}(\sin(\varphi)) \left[-m\bar{C}_{l,m}\sin(m\lambda) + m\bar{S}_{l,m}\cos(m\lambda)\right] \\
&= \frac{GM}{r} \sum_{l=0}^{l_{\max}} \sum_{m=0}^{l} \left(\frac{R_{\mathrm{E}}}{r}\right)^{l} \bar{P}_{l,m}(\sin(\varphi))m \left[\bar{S}_{l,m}\cos(m\lambda) - \bar{C}_{l,m}\sin(m\lambda)\right]
\end{aligned} \tag{3.69}$$

Mithilfe der Rotationsmatrix $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{L}}$ überführt man dann den Gravitationsbeschleunigungsvektor zuerst in das erdfeste Bezugssystem:

$$\begin{aligned}\nabla_{\mathrm{E}} V(r,\varphi,\lambda,\boldsymbol{p}) &= \boldsymbol{R}_{\mathrm{E}}^{\mathrm{L}}(\varphi,\lambda)\ \nabla_{\mathrm{L}} V(r,\varphi,\lambda,\boldsymbol{p}) = \left(\boldsymbol{R}_{\mathrm{L}}^{\mathrm{E}}(\varphi,\lambda)\right)^{\top}\ \nabla_{\mathrm{L}} V(r,\varphi,\lambda,\boldsymbol{p}) \\ &= \begin{pmatrix} \cos(\varphi)\cos(\lambda) & -\sin(\varphi)\cos(\lambda) & -\sin(\lambda) \\ \cos(\varphi)\sin(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\lambda) \\ \sin(\varphi) & \cos(\varphi) & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial V}{\partial r} \\ \frac{1}{r}\frac{\partial V}{\partial \varphi} \\ \frac{1}{r\cos(\varphi)}\frac{\partial V}{\partial \lambda} \end{pmatrix}. \end{aligned} \tag{3.70}$$

Anschließend erhält man durch eine weitere Transformation die endgültige Kräftefunktion bezüglich des raumfesten Systems:

$$\boldsymbol{f}_{\mathrm{K}}(t,\boldsymbol{X}(t),\boldsymbol{p}) = \boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}(t)\ \nabla_{\mathrm{E}} V(r,\varphi,\lambda,\boldsymbol{p}) \tag{3.71}$$

Dabei ist die zeitabhängige Transformationsmatrix $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}(t)$ vom geozentrisch erdfesten in das vereinbarte raumfeste System im Falle der Verarbeitung realer Daten normalerweise nach den IERS-Richtlinien (Petit und Luzum, 2010) zu bestimmen. In dieser Arbeit sind die Matrizen $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}$ (und $\boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}$) Bestandteil der gegebenen realen Daten, weshalb deren Bestimmung hier nicht näher erörtert wird und auch nicht in der eigenen Toolbox erfolgt.

Im lokalen Horizontsystem lässt sich auch für den (symmetrischen) Marussi-Tensor ein Ausdruck finden, der sich aus Funktionalen des Gravitationspotentials in terrestrischen, sphärischen Koordinaten zusammensetzt. Eine Herleitung hierfür geben z.B. Reed (1973, S. 31 ff.) sowie Scheinert (1996, S. 28). Der Marussi-Tensor ist demnach mit Bezug auf das hier gewählte lokale Horizontsystem gegeben durch

$$\boldsymbol{M}_{\mathrm{L}}(r,\varphi,\lambda,\boldsymbol{p}) = \begin{pmatrix} \frac{\partial^2 V}{\partial r^2} & \frac{1}{r}\frac{\partial^2 V}{\partial r\partial\varphi} - \frac{1}{r^2}\frac{\partial V}{\partial\varphi} & \frac{1}{r\cos(\varphi)}\frac{\partial^2 V}{\partial r\partial\lambda} - \frac{1}{r^2\cos(\varphi)}\frac{\partial V}{\partial\lambda} \\ & \frac{1}{r^2}\frac{\partial^2 V}{\partial\varphi^2} + \frac{1}{r}\frac{\partial V}{\partial r} & \frac{1}{r^2\cos(\varphi)}\frac{\partial^2 V}{\partial\varphi\partial\lambda} + \frac{\tan(\varphi)}{r^2\cos(\varphi)}\frac{\partial V}{\partial\lambda} \\ \text{(symmetrisch)} & & \frac{1}{r^2\cos^2(\varphi)}\frac{\partial^2 V}{\partial\lambda^2} - \frac{\tan(\varphi)}{r^2}\frac{\partial V}{\partial\varphi} + \frac{1}{r}\frac{\partial V}{\partial r} \end{pmatrix}. \tag{3.72}$$

Dabei ergeben sich die zweiten partiellen Ableitungen zu (vgl. Götzelmann, 2003, S. 43; Meyer, 2006, S. 104):

$$\frac{\partial^2 V}{\partial r^2} = \frac{GM}{r^3}\sum_{l=0}^{l_{\max}}\sum_{m=0}^{l}\left(\frac{R_{\mathrm{E}}}{r}\right)^l (l+1)(l+2)\bar{P}_{l,m}(\sin(\varphi))\left[\bar{C}_{l,m}\cos(m\lambda)+\bar{S}_{l,m}\sin(m\lambda)\right] \tag{3.73}$$

$$\frac{\partial^2 V}{\partial r\partial\varphi} = -\frac{GM}{r^2}\sum_{l=0}^{l_{\max}}\sum_{m=0}^{l}\left(\frac{R_{\mathrm{E}}}{r}\right)^l (l+1)\frac{\partial\bar{P}_{l,m}(\sin(\varphi))}{\partial\varphi}\left[\bar{C}_{l,m}\cos(m\lambda)+\bar{S}_{l,m}\sin(m\lambda)\right] \tag{3.74}$$

$$\frac{\partial^2 V}{\partial r\partial\lambda} = -\frac{GM}{r^2}\sum_{l=0}^{l_{\max}}\sum_{m=0}^{l}\left(\frac{R_{\mathrm{E}}}{r}\right)^l (l+1)\bar{P}_{l,m}(\sin(\varphi))m\left[\bar{S}_{l,m}\cos(m\lambda)-\bar{C}_{l,m}\sin(m\lambda)\right] \tag{3.75}$$

$$\frac{\partial^2 V}{\partial\varphi^2} = \frac{GM}{r}\sum_{l=0}^{l_{\max}}\sum_{m=0}^{l}\left(\frac{R_{\mathrm{E}}}{r}\right)^l \frac{\partial^2\bar{P}_{l,m}(\sin(\varphi))}{\partial\varphi^2}\left[\bar{C}_{l,m}\cos(m\lambda)+\bar{S}_{l,m}\sin(m\lambda)\right] \tag{3.76}$$

$$\frac{\partial^2 V}{\partial\varphi\partial\lambda} = \frac{GM}{r}\sum_{l=0}^{l_{\max}}\sum_{m=0}^{l}\left(\frac{R_{\mathrm{E}}}{r}\right)^l \frac{\partial\bar{P}_{l,m}(\sin(\varphi))}{\partial\varphi}m\left[\bar{S}_{l,m}\cos(m\lambda)-\bar{C}_{l,m}\sin(m\lambda)\right] \tag{3.77}$$

$$\frac{\partial^2 V}{\partial\lambda^2} = -\frac{GM}{r}\sum_{l=0}^{l_{\max}}\sum_{m=0}^{l}\left(\frac{R_{\mathrm{E}}}{r}\right)^l \bar{P}_{l,m}(\sin(\varphi))m^2\left[\bar{C}_{l,m}\cos(m\lambda)+\bar{S}_{l,m}\sin(m\lambda)\right] \tag{3.78}$$

Der Marussi-Tensor lässt sich nun durch die orthogonale Transformation

$$\boldsymbol{M}_{\mathrm{E}}(r,\varphi,\lambda,\boldsymbol{p}) = \boldsymbol{R}_{\mathrm{E}}^{\mathrm{L}}(\varphi,\lambda)\ \boldsymbol{M}_{\mathrm{L}}(r,\varphi,\lambda,\boldsymbol{p})\ \left(\boldsymbol{R}_{\mathrm{E}}^{\mathrm{L}}(\varphi,\lambda)\right)^{\top} \tag{3.79}$$

in das erdfeste Bezugssystem und anschließend in analoger Weise durch

$$\boldsymbol{M}(t,\boldsymbol{X}(t),\boldsymbol{p}) = \boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}(t)\ \boldsymbol{M}_{\mathrm{E}}(r,\varphi,\lambda,\boldsymbol{p})\ \left(\boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}}(t)\right)^{\top} \tag{3.80}$$

in das raumfeste Bezugssystem überführen.

Für obige partielle Ableitungen des Gravitationspotentials sind die vollständig normierten Legendre-Funktionen sowie die ersten und zweiten Ableitungen derselben zu bestimmen. Neben anderen möglichen Formeln verwendet man hierzu auf dem Computer üblicherweise die Rekursionsformeln, da sie vergleichsweise effizient hinsichtlich Rechenzeit und -genauigkeit sind. Derartige Vorschriften sind etwa in Scheinert (1996, S. 133) aufgezeigt.

Im Falle polnaher Berechnungspunkte kann es eventuell bei den obigen Ausdrücken für die Ableitungen des Gravitationspotentials zu Problemen kommen, da in den Rekursionsformeln für die Legendre-Funktionen und deren Ableitungen sowie in Gl. (3.70) und (3.72) singuläre Terme wie z.B. $\cos^{-1}(\varphi)$ auftreten. Verschiedene Autoren (z.B. Petrovskaya und Vershkov, 2006; Eshagh, 2010) haben nicht-singuläre Formeln hergeleitet, die auch für polnahe Bahnen die (zweiten) Ableitungen zuverlässig liefern.

Der prinzipielle Berechnungsablauf mit den in diesem Abschnitt behandelten Formeln kann wie folgt zusammengefasst werden: zuerst überführt man den im raumfesten System gegebenen Positionsvektor des Satelliten (Anfangswerte) mithilfe der Matrix $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}(t)$ in erdfeste, kartesische und anschließend in erdfeste sphärische Koordinaten[1]. Danach bestimmt man mithilfe obiger Formeln die Gravitationsbeschleunigung und den Marussi-Tensor im lokalen Horizontsystem, transformiert diese in das quasi-inertiale System und stellt schließlich damit das zu integrierende DGL-System auf.

3.2.5.2 Kräftemodell für die Satellitenbewegung im Gravitationsfeld eines abgeplatteten Erdkörpers

Dieses Kräftemodell soll neben dem Zentralfeld nur die Erdabplattung berücksichtigen. Es stellt einen Spezialfall des vorhergehenden Kräftemodells dar, denn man erhielte ein gleichartiges Modell auch, indem man im vorigen Modell $l_{\max} = 2$ annimmt. Hier können allerdings die gesuchten dynamischen Größen direkt im raumfesten System formuliert werden, d.h. es sind keine Rotationen zwischen den Koordinatensystemen nötig.

Zunächst wertet man dazu die Kugelfunktionsentwicklung des Gravitationspotentials (hier genügt die unnormierte Variante) bis Grad $l = 2$ und Ordnung $m = 0$ aus, woraus die Beziehung

$$\begin{aligned} V_2(r,\varphi,\boldsymbol{p}) &= \frac{GM}{R_{\mathrm{E}}}\cdot\left(\frac{R_{\mathrm{E}}}{r}\right) + \frac{GM}{R_{\mathrm{E}}}\cdot\left(\frac{R_{\mathrm{E}}}{r}\right)^3\cdot P_{2,0}(\sin(\varphi))\cdot C_{2,0} \\ &= \frac{GM}{r} + \underbrace{\frac{GM}{R_{\mathrm{E}}}\cdot\left(\frac{R_{\mathrm{E}}}{r}\right)^3\cdot\frac{1}{2}(3\sin^2(\varphi)-1)\cdot C_{2,0}}_{R_{2,0}(r,\varphi,\boldsymbol{p})} \end{aligned} \tag{3.81}$$

mit dem J_2-Störpotential $R_{2,0}$ folgt. Hierin entfallen sämtliche Terme ersten Grades, da, wie bei den anderen Kräftemodellen auch, angenommen wird, dass das zugrundeliegende erdfeste Koordinatensystem geozentrisch gelagert ist und die Koordinatenachsen des Bezugssystems mit

1 Dies erklärt die Abhängigkeit der Kraftfunktion $\boldsymbol{f}_{\mathrm{K}}$ und des Tensors $\boldsymbol{M}$ vom raumfesten Positionsvektor $\boldsymbol{X}(t)$.

den Hauptträgheitsachsen der Erde zusammenfallen. Die einzelnen Variablen in Gl. (3.81) haben folgende Bedeutung:

r Länge des Ortsvektors des Satelliten bzw. dessen radialer Abstand zum Geozentrum.

φ geozentrische Breite des Satelliten.

$P_{2,0}$ zugeordnete, nicht normierte Legendre-Funktion, $P_{2,0}(\sin(\varphi)) = \frac{1}{2}(3\sin^2(\varphi) - 1)$.

GM Geozentrische Gravitationskonstante, hier $GM = 3986004.415 \cdot 10^{14}\,\mathrm{m^3/s^2}$

R_E Äquatorradius, hier $R_\mathrm{E} = a_\mathrm{E} = 6378136.3\,\mathrm{m}$

$C_{2,0}$ unnormierter Kugelfunktionskoeffizient, der den Einfluss der Erdabplattung auf die Massenverteilung repräsentiert; z.B. bei EGM96: $J_2 = -C_{2,0} \approx 1.08262668 \cdot 10^{-3}$.

In diesem Modell beinhalte $\boldsymbol{p}$ die drei zuletzt genannten Größen.

Indem man Präzession, Nutation und Polbewegung vernachlässigt, kann man für das J_2-Störpotential einen (approximierten) Ausdruck direkt im raumfesten System finden. Mit dieser Annahme fällt die z-Achse des erdfesten mit der Z-Achse des quasi-inertialen Systems zusammen, da als Ursprung beider Systeme das Geozentrum vereinbart wurde. Somit gilt

$$\sin(\varphi) = \frac{z}{r} = \frac{Z}{r}.$$

Da die Bezugssysteme außerdem keinen Maßstabsunterschied zueinander aufweisen, ist die Länge des Ortsvektors

$$r = \sqrt{x^2 + y^2 + z^2} = \sqrt{X^2 + Y^2 + Z^2}$$

in beiden Systemen identisch und das Störpotential der Erdabplattung $R_{2,0}$ lässt sich direkt im quasi-inertialen System angeben zu (vgl. Sneeuw, 2006, S. 86):

$$\begin{aligned} R_{2,0}(\boldsymbol{X}(t), \boldsymbol{p}) &= \frac{GM}{R_\mathrm{E}} \cdot \left(\frac{R_\mathrm{E}}{r}\right)^3 \cdot \frac{1}{2}(3\sin^2(\varphi) - 1) \cdot C_{2,0} \\ &= \frac{GM}{R_\mathrm{E}} \cdot \left(\frac{R_\mathrm{E}}{r}\right)^3 \cdot \frac{1}{2r^2}(3Z^2 - r^2) \cdot C_{2,0} \\ &= \frac{1}{2} \cdot \frac{GM}{R_\mathrm{E}} \cdot \frac{R_\mathrm{E}^3}{r^5} \cdot C_{2,0} \cdot (2Z^2 - X^2 - Y^2). \end{aligned} \tag{3.82}$$

Mit dem in Anhang B.3 (Seite 148 ff.) hergeleiteten Gradienten des Störpotentials $R_{2,0}$ folgt für die Kräftefunktion in diesem Modell die Beziehung

$$\begin{aligned} \boldsymbol{f}_\mathrm{K}(\boldsymbol{X}(t), \boldsymbol{p}) &= \nabla V_2(\boldsymbol{X}(t), \boldsymbol{p}) = \nabla \frac{GM}{r} + \nabla R_{2,0}(\boldsymbol{X}(t), \boldsymbol{p}) \\ &= -\frac{GM}{r^3}\boldsymbol{X} + \begin{pmatrix} -\frac{3}{2} \cdot GM \cdot \left(\frac{R_\mathrm{E}}{r}\right)^2 \cdot C_{2,0} \cdot \frac{X}{r^3}\left(5 \cdot \frac{Z^2}{r^2} - 1\right) \\ -\frac{3}{2} \cdot GM \cdot \left(\frac{R_\mathrm{E}}{r}\right)^2 \cdot C_{2,0} \cdot \frac{Y}{r^3}\left(5 \cdot \frac{Z^2}{r^2} - 1\right) \\ -\frac{3}{2} \cdot GM \cdot \left(\frac{R_\mathrm{E}}{r}\right)^2 \cdot C_{2,0} \cdot \frac{Z}{r^3}\left(5 \cdot \frac{Z^2}{r^2} - 3\right) \end{pmatrix}. \end{aligned} \tag{3.83}$$

Für den Marussi-Tensor der J_2-gestörten Bahn werden nun die zweiten Ableitungen des Potentials

$$V_2(\boldsymbol{X}(t), \boldsymbol{p}) = \frac{GM}{r} + \frac{1}{2} \cdot \frac{GM}{R_\mathrm{E}} \cdot \frac{R_\mathrm{E}^3}{r^5} \cdot C_{2,0} \cdot (2Z^2 - X^2 - Y^2)$$

nach den kartesischen Koordinaten benötigt. Durch nochmaliges Differenzieren der in Gl. (3.83) gegebenen ersten Ableitungen nach den kartesischen Koordinaten entsprechend Gl. (3.55) erhält man folgende Einträge des J_2-Marussi-Tensors:

$$\begin{aligned} M_{11} = \frac{\partial^2 V_{2,0}}{\partial X^2} = -\frac{GM}{r^3} + \frac{GM}{r^5} \cdot \left(\frac{3}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} + 3 \cdot X^2 \right) \\ - \frac{GM}{r^7} \cdot \frac{15}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \left(X^2 + Z^2 \right) \\ + \frac{GM}{r^9} \cdot \left(\frac{105}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot X^2 \cdot Z^2 \right) \end{aligned} \quad (3.84)$$

$$\begin{aligned} M_{12} = \frac{\partial^2 V_{2,0}}{\partial X \partial Y} = \frac{3 \cdot GM}{r^5} \cdot X \cdot Y - \frac{GM}{r^7} \cdot \left(\frac{15}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot X \cdot Y \right) \\ + \frac{GM}{r^9} \cdot \left(\frac{105}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot X \cdot Z^2 \cdot Y \right) \end{aligned} \quad (3.85)$$

$$\begin{aligned} M_{13} = \frac{\partial^2 V_{2,0}}{\partial X \partial Z} = \frac{3 \cdot GM}{r^5} \cdot X \cdot Z - \frac{GM}{r^7} \cdot \left(\frac{45}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot X \cdot Z \right) \\ + \frac{GM}{r^9} \cdot \left(\frac{105}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot X \cdot Z^3 \right) \end{aligned} \quad (3.86)$$

$$\begin{aligned} M_{22} = \frac{\partial^2 V_{2,0}}{\partial Y^2} = -\frac{GM}{r^3} + \frac{GM}{r^5} \cdot \left(\frac{3}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} + 3 \cdot Y^2 \right) \\ - \frac{GM}{r^7} \cdot \frac{15}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \left(Y^2 + Z^2 \right) \\ + \frac{GM}{r^9} \cdot \left(\frac{105}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot Y^2 \cdot Z^2 \right) \end{aligned} \quad (3.87)$$

$$\begin{aligned} M_{23} = \frac{\partial^2 V_{2,0}}{\partial Y \partial Z} = \frac{3 \cdot GM}{r^5} \cdot Y \cdot Z - \frac{GM}{r^7} \cdot \left(\frac{45}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot Y \cdot Z \right) \\ + \frac{GM}{r^9} \cdot \left(\frac{105}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot Y \cdot Z^3 \right) \end{aligned} \quad (3.88)$$

$$\begin{aligned} M_{33} = \frac{\partial^2 V_{2,0}}{\partial Z^2} = -\frac{GM}{r^3} + \frac{GM}{r^5} \cdot \left(\frac{9}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} + 3 \cdot Z^2 \right) - \frac{GM}{r^7} \cdot \left(45 \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot Z^2 \right) \\ + \frac{GM}{r^9} \cdot \left(\frac{105}{2} \cdot R_\mathrm{E}^2 \cdot C_{2,0} \cdot Z^4 \right) \end{aligned} \quad (3.89)$$

Somit ist das nach jedem Integrationsschritt neu auszuwertende DGL-System für den J_2-gestörten Fall formelmäßig vollständig festgelegt.

3.2.5.3 Numerische Lösung des ungestörten Zweikörperproblems

Das klassische Zweikörperproblem, für das bekanntlich sowohl numerische als auch analytische Lösungen (s. nächster Abschnitt) existieren, wird hier als Kräftemodell im Grunde nur deshalb betrachtet, um beide Lösungsarten vergleichen zu können bzw. um später die Genauigkeit der numerischen Integration zu überprüfen. Wie schon im letzten Kapitel festgestellt wurde, sind Keplerorbits jedoch zur Verdichtung realer Orbits selbst nicht zu empfehlen. Denn bereits der um den Einfluss der Erdabplattung erweiterte Orbitintegrator ermöglicht eine genauere Prädiktion bei vergleichbaren Rechenzeiten.

Im Keplerfall ist das Gravitationsfeld mit dem Zentralfeld einer homogenen Kugel mit Radius r identisch. Demzufolge besteht das kugelsymmetrische Gravitationspotential nur aus dem Zentral- bzw. Keplerterm

$$V_0(\boldsymbol{X}(t), \boldsymbol{p}) = \frac{GM}{r} \quad \text{mit} \quad r = \sqrt{X^2(t) + Y^2(t) + Z^2(t)} \quad \text{und} \quad \boldsymbol{p} = \{GM\}. \tag{3.90}$$

Die Kräftefunktion ist hier der Gradient des Zentralterms. Unter Ausnutzung der Beziehungen

$$\begin{aligned} \frac{\partial r}{\partial X} &= \frac{\partial}{\partial X}\sqrt{X^2+Y^2+Z^2} = \frac{2X}{2\sqrt{X^2+Y^2+Z^2}} = \frac{X}{r}, \\ &\Longrightarrow \quad \frac{\partial r}{\partial Y} = \frac{Y}{r}, \quad \frac{\partial r}{\partial Z} = \frac{Z}{r} \end{aligned} \tag{3.91}$$

folgt für die ersten partiellen Ableitungen des Zentralterms

$$\frac{\partial V_0}{\partial X} = -\frac{GM}{r^2}\frac{\partial r}{\partial X} \overset{(3.91)}{=} -\frac{GM}{r^3} \cdot X(t) \tag{3.92}$$

$$\frac{\partial V_0}{\partial Y} = -\frac{GM}{r^2}\frac{\partial r}{\partial Y} \overset{(3.91)}{=} -\frac{GM}{r^3} \cdot Y(t) \tag{3.93}$$

$$\frac{\partial V_0}{\partial Z} = -\frac{GM}{r^2}\frac{\partial r}{\partial Z} \overset{(3.91)}{=} -\frac{GM}{r^3} \cdot Z(t). \tag{3.94}$$

Damit lauten die ersten sechs Gleichungen des zu integrierenden DGL-Systems

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \boldsymbol{X}(t) \\ \dot{\boldsymbol{X}}(t) \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{X}}(t) \\ \boldsymbol{f}_{\mathrm{K}}(t, \boldsymbol{X}(t), \boldsymbol{p}) \end{pmatrix} \overset{\text{hier}}{=} \begin{pmatrix} \dot{\boldsymbol{X}}(t) \\ -\frac{GM}{r^3}\boldsymbol{X}(t) \end{pmatrix}. \tag{3.95}$$

Für die restlichen DGLen benötigt man (bei allen Methoden `'ric'`, `'lin'` und `'const'`) wiederum den Marussi-Tensor. Dazu sind die Gleichungen (3.92) bis (3.94) nochmals nach den kartesischen raumfesten Koordinaten zu differenzieren:

$$M_{11} = \frac{\partial^2 V_0}{\partial X \partial X} = \frac{3 \cdot GM \cdot X}{r^4} \cdot \frac{\partial r}{\partial X} - \frac{GM}{r^3} \overset{(3.91)}{=} \frac{3 \cdot GM \cdot X^2}{r^5} - \frac{GM}{r^3} \tag{3.96}$$

$$M_{12} = \frac{\partial^2 V_0}{\partial X \partial Y} = \frac{3 \cdot GM \cdot X}{r^4} \cdot \frac{\partial r}{\partial Y} \overset{(3.91)}{=} \frac{3 \cdot GM \cdot XY}{r^5} \tag{3.97}$$

$$M_{13} = \frac{\partial^2 V_0}{\partial X \partial Z} = \frac{3 \cdot GM \cdot X}{r^4} \cdot \frac{\partial r}{\partial Z} \overset{(3.91)}{=} \frac{3 \cdot GM \cdot XZ}{r^5} \tag{3.98}$$

$$M_{22} = \frac{\partial^2 V_0}{\partial Y \partial Y} = \frac{3 \cdot GM \cdot Y}{r^4} \cdot \frac{\partial r}{\partial Y} - \frac{GM}{r^3} \overset{(3.91)}{=} \frac{3 \cdot GM \cdot Y^2}{r^5} - \frac{GM}{r^3} \tag{3.99}$$

$$M_{23} = \frac{\partial^2 V_0}{\partial Y \partial Z} = \frac{3 \cdot GM \cdot Y}{r^4} \cdot \frac{\partial r}{\partial Z} \overset{(3.91)}{=} \frac{3 \cdot GM \cdot YZ}{r^5} \tag{3.100}$$

$$M_{33} = \frac{\partial^2 V_0}{\partial Z \partial Z} = \frac{3 \cdot GM \cdot Z}{r^4} \cdot \frac{\partial r}{\partial Z} - \frac{GM}{r^3} \overset{(3.91)}{=} \frac{3 \cdot GM \cdot Z^2}{r^5} - \frac{GM}{r^3} \tag{3.101}$$

Somit lautet der Marussi-Tensor nun im Keplerfall

$$\boldsymbol{M}(\bar{\boldsymbol{s}}(t), \boldsymbol{p}) = \frac{GM}{r^5} \begin{pmatrix} 3X^2 - r^2 & 3XY & 3XZ \\ 3YX & 3Y^2 - r^2 & 3YZ \\ 3ZX & 3ZY & 3Z^2 - r^2 \end{pmatrix}. \tag{3.102}$$

3.2.5.4 Analytische Lösung des ungestörten Zweikörperproblems

Unter den analytischen Lösungsmöglichkeiten für das ungestörte Zweikörperproblem gibt es neben der bereits im zweiten Kapitel erläuterten Ephemeridenrechnung (Seite 19) auch eine sogenannte Lagrange/Gibbs-Lösung (siehe Schaub und Junkins, 2003, Kap. 9.5; Escobal, 1976, S. 112). Dabei werden zukünftige Bahnvektoren des Keplerorbits als Linearkombination der Anfangsbedingungen[1] dargestellt:

$$\bar{\boldsymbol{s}}(t) = \begin{pmatrix} \boldsymbol{X}(t) \\ \dot{\boldsymbol{X}}(t) \end{pmatrix} = \begin{pmatrix} F \cdot \mathbf{I}_{(3\times3)} & G \cdot \mathbf{I}_{(3\times3)} \\ \dot{F} \cdot \mathbf{I}_{(3\times3)} & \dot{G} \cdot \mathbf{I}_{(3\times3)} \end{pmatrix} \hat{\boldsymbol{s}}_0. \tag{3.103}$$

Diese Darstellung ist möglich, weil bei der Keplerbewegung alle Orts- und Geschwindigkeitsvektoren in einer Bahnebene liegen (Kepler-Ellipse), welche durch die sechs Anfangsbedingungen $\boldsymbol{X}_0, \dot{\boldsymbol{X}}_0$ (oder alternativ die zugehörigen Keplerelemente) eindeutig bezüglich des raumfesten geozentrischen Systems festgelegt ist.

Nach Escobal (1976, S. 112) sowie Schaub und Junkins (2003, Kap. 9.5) können die darin enthaltenen, zeitvariablen skalaren Koeffizienten F und G und deren Ableitungen entweder in Form von Reihenentwicklungen oder aber geschlossen gefunden werden zu

$$F = 1 - \frac{a}{r_0}(1 - \cos(\Delta E)) \qquad G = \Delta t + \sqrt{\frac{a^3}{GM}} \cdot (\sin(\Delta E) - \Delta E) \tag{3.104}$$

$$\dot{F} = -\frac{\sqrt{GM \cdot a}}{r \cdot r_0} \cdot \sin(\Delta E) \qquad \dot{G} = 1 + \frac{a}{r} \cdot (\cos(\Delta E) - 1). \tag{3.105}$$

Hierin bezeichnet $\Delta t = t_i - t_0 \mathrel{\widehat{=}} i \cdot \delta t$ die Zeitdifferenz zum Anfangszeitpunkt und $\Delta E = E(t_i) - E_0$ die Differenz zwischen der momentanen exzentrischen Anomalie und derjenigen am Startzeitpunkt. r_0 ist die Länge bzw. Norm des Ortsvektors am Anfangszeitpunkt

$$r_0 = \|\boldsymbol{X}_0\| = \sqrt{X_0^2 + Y_0^2 + Z_0^2} \tag{3.106}$$

Für den momentanen Bahnradius gilt den genannten Quellen zufolge

$$r = a\,(1 - e\cos(E_0)\cos(\Delta E) + e\sin(E_0)\sin(\Delta E)). \tag{3.107}$$

Um die zeitvariable Matrix in Gl. (3.105) zu bestimmen und damit einen zukünftigen Bahnvektor bezüglich des raumfesten geozentrischen Systems zu erhalten, wird in den obigen Beziehungen neben den Keplerelementen des Anfangszeitpunkts auch die exzentrische Anomalie $E(t)$ am gewünschten zukünftigen Zeitpunkt t benötigt. Diese erhält man z.B. durch lineare Prädiktion der mittleren Anomalie gemäß

$$M(t_i) = M_0 + n \cdot i \cdot \delta t \quad \text{mit} \quad n = \sqrt{GM/a^3}$$

1 Die auf den Anfangszeitpunkt t_k bezogenen Größen, u.a. auch der gegebene Startvektor, sind in diesem Abschnitt mit dem Index 0 versehen.

und iterative Auflösung der Keplergleichung $M(t) = E(t) - e \cdot \sin(E(t))$ nach $E(t)$ (s. z.B. Curtis, 2010, Algorithmus 3.1, S. 163).

Richard H. Battin hat auf Grundlage dieser Lagrange/Gibbs-Lösung geschlossen analytische Beziehungen hergeleitet für die (im Keplerfall) zur Fehlerfortpflanzung benötigte Transitionsmatrix, vgl. Battin (1999); Schaub und Junkins (2003, Kap. 12.4). Er definiert einen weiteren skalaren Koeffizienten

$$C = a \cdot \sqrt{\frac{a^3}{GM}} \cdot \big(3\sin(\Delta E) - (2 + \cos(\Delta E))\Delta E\big) - \Delta t \cdot a\,(1 - \cos(\Delta E)) \tag{3.108}$$

und setzt die Transitionsmatrix

$$\boldsymbol{\Phi}(t, t_0) = \frac{\partial \bar{\boldsymbol{s}}(t)}{\partial \hat{\boldsymbol{s}}_0} \overset{\text{hier}}{=} \begin{pmatrix} \boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} \\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22} \end{pmatrix} \tag{3.109}$$

aus den folgenden vier (3×3)-Submatrizen zusammen:

$$\boldsymbol{\Phi}_{11} = \frac{r}{GM}\delta\dot{\boldsymbol{X}}\delta\dot{\boldsymbol{X}}^\top + \frac{1}{r_0^3}\Big(r_0(1-F)\boldsymbol{X}\boldsymbol{X}_0^\top + C\dot{\boldsymbol{X}}\boldsymbol{X}_0^\top\Big) + F \cdot \mathbf{I}_{(3\times3)} \tag{3.110}$$

$$\boldsymbol{\Phi}_{12} = \frac{r_0}{GM}(1-F)(\delta\boldsymbol{X}\dot{\boldsymbol{X}}_0^\top - \delta\dot{\boldsymbol{X}}\boldsymbol{X}_0^\top) + \frac{C}{GM}\dot{\boldsymbol{X}}\dot{\boldsymbol{X}}_0^\top + G \cdot \mathbf{I}_{(3\times3)} \tag{3.111}$$

$$\begin{aligned}\boldsymbol{\Phi}_{21} = & -\frac{1}{r_0^2}\delta\dot{\boldsymbol{X}}\boldsymbol{X}_0^\top - \frac{1}{r^2}\boldsymbol{X}\delta\dot{\boldsymbol{X}}^\top - \frac{GM \cdot C}{r^3 \cdot r_0^3}\boldsymbol{X}\boldsymbol{X}_0^\top \\ & + \dot{F}\Big(\mathbf{I}_{(3\times3)} - \frac{1}{r^2}\boldsymbol{X}\boldsymbol{X}^\top + \frac{1}{GM \cdot r}(\boldsymbol{X}\dot{\boldsymbol{X}}^\top - \dot{\boldsymbol{X}}\boldsymbol{X}^\top)\boldsymbol{X}\delta\dot{\boldsymbol{X}}^\top\Big)\end{aligned} \tag{3.112}$$

$$\boldsymbol{\Phi}_{22} = \frac{r_0}{GM}\delta\dot{\boldsymbol{X}}\delta\dot{\boldsymbol{X}}^\top + \frac{1}{r^3}\Big(r_0(1-F)\boldsymbol{X}\boldsymbol{X}_0^\top - C\boldsymbol{X}\dot{\boldsymbol{X}}_0^\top\Big) + \dot{G} \cdot \mathbf{I}_{(3\times3)} \tag{3.113}$$

Dabei ist $\delta\boldsymbol{X} = \boldsymbol{X}(t) - \boldsymbol{X}_0$ der Vektor der Positionsdifferenzen und $\delta\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}(t) - \dot{\boldsymbol{X}}_0$ der Vektor der Geschwindigkeitsdifferenzen zwischen dem nach Gl. (3.103) prädizierten Bahnvektor und dem gegebenen Anfangsvektor. Die Produkte $\delta\dot{\boldsymbol{X}}\delta\dot{\boldsymbol{X}}^\top$ usw. ergeben hier jeweils eine (3×3)-Matrix.

Man bedenke, dass die Transitionsmatrix $\boldsymbol{\Phi}(t, t_0)$ zur Fortschreibung der Schätzfehlerkovarianzen bzw. der Zustandsfehler entsprechend dem linearisierten Fehlersystemmodell (3.39) dient und *nicht* gleichbedeutend ist mit der Matrix in Gleichung (3.103), welche den Ausgangszustand selbst auf einen späteren Zeitpunkt $t > t_0$ abbildet.

Damit stehen nun alle Formeln zur Verfügung, um im ungestörten Zweikörper-Kräftemodell einerseits die Zustandsvektoren und andererseits die Transitionsmatrizen (Methoden `'lin'` und `'const'`) vollständig analytisch, d.h. ohne numerische Integration, fortzuschreiben.

3.2.6 Korrektur der prädizierten Größen durch Beobachtungen

Im vorigen Abschnitt wurden die Kräftemodelle vorgestellt, die in der Toolbox verwendet werden können, um die Systemzustände und die zugehörigen Kovarianzen ausgehend von entsprechenden Anfangswerten zu prädizieren. Wir nehmen an, dass nun der Zustandsvektor $\bar{\boldsymbol{s}}_1$ und dessen Schätzfehlerkovarianzmatrix $\bar{\boldsymbol{P}}_1$ zum Zeitpunkt der ersten Beobachtung vorliegen. Abb. 3.5 veranschaulicht den nun durchzuführenden Korrekturschritt im Zeitbereich. Die Kovarianzmatrizen sind dabei exemplarisch durch Fehlerellipsen angedeutet, woran ersichtlich ist, dass sich bei der Korrektur mithilfe von geeigneten Beobachtungen ($\boldsymbol{b}_1$, Kovarianzen $\boldsymbol{R}_1$) ein verbesserter, d.h. statistisch betrachtet genauerer Systemzustand ($\hat{\boldsymbol{s}}_1$, Kovarianzen $\widehat{\boldsymbol{P}}_1$) ermitteln lässt, der dann als Ausgangszustand für die nächste Prädiktion dient.

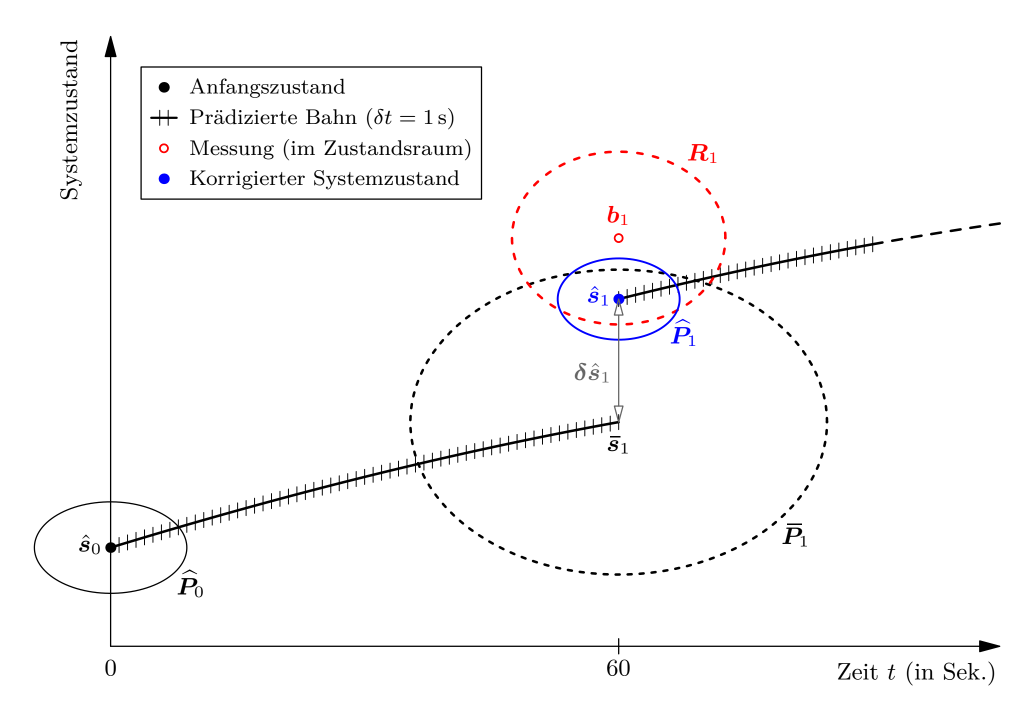


Abb. 3.5: Zeitlicher Ablauf des Filters bei Verwendung der Funktion `ekf_orbit`

Um die Korrektur mittels der in Tafel 3.2 gegebenen Formeln durchführen zu können, muss neben den bereits genannten Größen einerseits die Funktion $\boldsymbol{h}$ und andererseits die Messmatrix $\boldsymbol{H}$ bekannt sein. Beide sind stets von der jeweiligen Beobachtung abhängig, also individuell analytisch herzuleiten. In der Toolbox wurden zwei unterschiedliche Filtervarianten zur Orbitverdichtung realisiert, die sich im Hinblick auf die eingehenden Beobachtungen unterscheiden:

- Die erste Variante (Funktion `ekf_orbit`) führt die Orbitverdichtung anhand *eines* (beliebigen) Satellitenorbits durch, der als Zeitreihe mit zugehörigen Kovarianzen im raumfesten System vorliegen muss. Dabei wird für die Korrektur also einzig der entsprechende Zustandsvektor des gegebenen Orbits als Beobachtung verwendet.
- Die andere Variante (`ekf_orbit2`) ist hingegen speziell für GRACE ausgelegt. Hier fließen bei der Korrektur die Bahnvektoren von zwei gegebenen Satellitenorbits und darüberhinaus auch noch die K-Band-Messungen $\dot{\rho}$ mit ein.

3.2.6.1 Korrekturschritt für einen einzelnen Orbit

Nachfolgend werden die benötigten Größen für den Korrekturschritt der ersten Filtervariante (Funktion `ekf_orbit`) spezifiziert. Da nur ein einzelner Orbit prozessiert wird, ist der Zustandsvektor

gegeben durch

$$\boldsymbol{s} = (\boldsymbol{X},\ \dot{\boldsymbol{X}})^\top = (X,\ Y,\ Z,\ \dot{X},\ \dot{Y},\ \dot{Z})^\top.$$

Die allgemeine nicht-lineare Beobachtungsgleichung lautet gemäß Gl. (3.24)

$$\boldsymbol{b}_k = \boldsymbol{h}(\boldsymbol{s}_k) + \boldsymbol{v}_k \quad \text{mit} \quad \boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}_k),$$

worin $\boldsymbol{h}$ den (nicht-linearen) funktionalen Zusammenhang zwischen den Beobachtungen und den Komponenten des Zustandsvektors beschreibt. Für die Beobachtungsgleichung gilt hier der lineare Zusammenhang

$$\begin{pmatrix} \boldsymbol{X}_{\text{nav}}(t_k) \\ \dot{\boldsymbol{X}}_{\text{nav}}(t_k) \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}(t_k) \\ \dot{\boldsymbol{X}}(t_k) \end{pmatrix} + \boldsymbol{v}_k. \tag{3.114}$$

wobei der Positions- und Geschwindigkeitsvektor des gegebenen bzw. beobachteten Orbits auf der linken Seite mit dem Index „nav" versehen ist, da in dieser Arbeit der gegebene Orbit gerade die GRACE-Navigationslösung ist.

Folglich gilt für die prädizierte Beobachtung $\boldsymbol{h}(\bar{\boldsymbol{s}}_k) = \bar{\boldsymbol{s}}_k$ und die Messmatrix ist einfach eine (6×6)-Einheitsmatrix:

$$\boldsymbol{H}(\bar{\boldsymbol{s}}(t_k)) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\bar{\boldsymbol{s}}(t_k)} = \left.\begin{pmatrix} \frac{\partial X}{\partial X} & \cdots & \frac{\partial X}{\partial \dot{Z}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \dot{Z}}{\partial X} & \cdots & \frac{\partial \dot{Z}}{\partial \dot{Z}} \end{pmatrix}\right|_{\bar{\boldsymbol{s}}(t_k)} = \mathbf{I}_{(6\times 6)} \tag{3.115}$$

In der zur Verfügung gestellten Navigationslösung sind zu jedem Zustandsvektor zwölf Fehlermaße (Standardabweichungen und Korrelationen) gegeben[1]. Für die Einträge der Kovarianzmatrix der Beobachtungen gilt generell

$$\boldsymbol{R}_{[ij]} = \text{cov}(\boldsymbol{b}_{[i]}, \boldsymbol{b}_{[j]}). \tag{3.116}$$

Somit besteht hier die Matrix des Messrauschens $\boldsymbol{v}_k$ aus zwei (3×3)-Submatrizen, die wie folgt mit den quadrierten gegebenen zwölf Fehlermaßen zu besetzen sind:

$$\boldsymbol{R}_k = \begin{pmatrix} \begin{matrix} \sigma^2_{X_\text{nav}} & \sigma_{X_\text{nav}Y_\text{nav}} & \sigma_{X_\text{nav}Z_\text{nav}} \\ & \sigma^2_{Y_\text{nav}} & \sigma_{Y_\text{nav}Z_\text{nav}} \\ \text{(sym.)} & & \sigma^2_{Z_\text{nav}} \end{matrix} & \mathbf{0} \\ \mathbf{0} & \begin{matrix} \sigma^2_{\dot{X}_\text{nav}} & \sigma_{\dot{X}_\text{nav}\dot{Y}_\text{nav}} & \sigma_{\dot{X}_\text{nav}\dot{Z}_\text{nav}} \\ & \sigma^2_{\dot{Y}_\text{nav}} & \sigma_{\dot{Y}_\text{nav}\dot{Z}_\text{nav}} \\ \text{(sym.)} & & \sigma^2_{\dot{Z}_\text{nav}} \end{matrix} \end{pmatrix} \tag{3.117}$$

Damit sind alle Größen formelmäßig bekannt, um in dieser Filtervariante nun die Aufdatierung nach den allgemeinen Gleichungen (Schritt 2 in Tafel 3.2, Seite 36) durchführen zu können.

1 Siehe bzgl. des Datenformats der Navigationslösungen auch Abschn. 4.1.1.

3.2.6.2 Korrektur im Falle zweier GRACE-Orbits mit eingehender K-Band-Messung $\dot{\rho}$

Im Falle von GRACE bietet es sich an, neben den beiden GRACE-Navigationslösungen zudem die hochgenauen K-Band-Messungen als ‚stützende‘[1] Beobachtungen zu verwenden.

Da die Phasenmehrdeutigkeiten der gemessenen Relativentfernungen ρ im zugrundeliegenden Datensatz KBR1B ungelöst sind und demnach die Werte einen unbekannten Offset aufweisen, wurde hier nur die Relativgeschwindigkeit $\dot{\rho}$ als Beobachtungsgröße für den Korrekturschritt gewählt. Diese hängt von den Orts- und Geschwindigkeitsvektoren *beider* Satelliten ab, weshalb der prädizierte Systemzustand *nur* für das Messungsupdate erweitert wird zu

$$\boldsymbol{s} = (\boldsymbol{X}_1,\ \dot{\boldsymbol{X}}_1,\ \boldsymbol{X}_2,\ \dot{\boldsymbol{X}}_2)^{\top}_{(12\times 1)}. \tag{3.118}$$

Die bereitgestellten K-Band-Daten haben ein zeitliches Intervall (‚Sampling‘) von 5 Sekunden, während die Navigationslösung alle 60 Sekunden vorliegt. Daher sind in dieser Filtervariante (`ekf_orbit2`) zwei unterschiedliche Messungsupdates erforderlich, wie in Abb. 3.6 dargestellt.

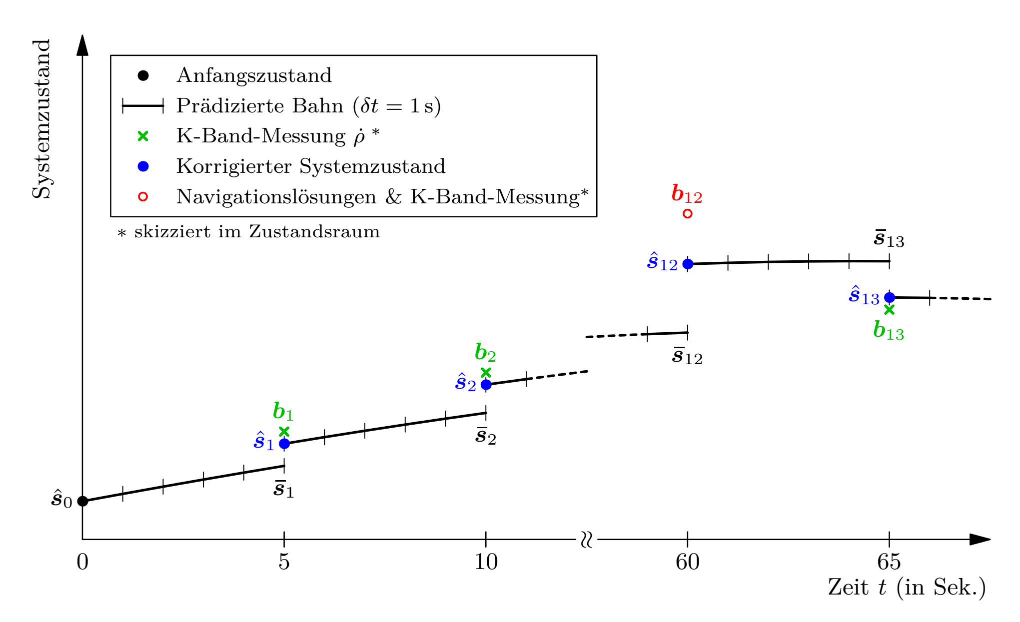


Abb. 3.6: Zeitlicher Ablauf des Filters bei Verwendung der Funktion `ekf_orbit2`

Betrachten wir zunächst die Korrektur, die mithilfe der K-Band-Beobachtung $\dot{\rho}$ alle 5 Sekunden stattfinden muss. Da diese Beobachtung nur ein skalarer Wert ist, reduziert sich die allgemeine vektorielle Beobachtungsgleichung zu einer nicht-linearen skalaren Gleichung:

$$\boldsymbol{b}_k = \boldsymbol{h}(\boldsymbol{s}_k) + \boldsymbol{v}_k \quad \Longrightarrow \quad \dot{\rho}_{\text{gem}}(t_k) = h(\boldsymbol{s}_k) + v_k \quad \text{mit} \quad v_k \sim \mathcal{N}(0, R_k). \tag{3.119}$$

Hierin ist nun der nicht-lineare Zusammenhang $h(\boldsymbol{s})$ zwischen der gemessenen Relativgeschwindigkeit $\dot{\rho}_{\text{gem}}$ und dem Zustandsvektor $\boldsymbol{s}$ anzugeben. Mithilfe Gleichung (2.2) und (2.3) erhalten wir

$$h(\boldsymbol{s}) = \dot{\rho} = \langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle \overset{(2.2)}{=} \frac{1}{\|\boldsymbol{X}_{12}\|} \langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{X}_{12} \rangle \tag{3.120}$$

mit den Differenzvektoren

$$\boldsymbol{X}_{12} = \boldsymbol{X}_2 - \boldsymbol{X}_1 \quad \text{und} \quad \dot{\boldsymbol{X}}_{12} = \dot{\boldsymbol{X}}_2 - \dot{\boldsymbol{X}}_1. \tag{3.121}$$

1 Aufgrund der überlegenen Genauigkeit der K-Band-Messungen haben diese beim Korrekturschritt beinahe die Wirkung einer Zwangsbedingung an die eingehenden Bahnvektoren.

Nach diesen Gleichungen ist beim K-Band-Update die prädizierte/erwartete Beobachtung $\boldsymbol{h}(\bar{\boldsymbol{s}}_k)$ aus den prädizierten Zustandsvektoren beider Satelliten zu ermitteln.

Die zugehörige Messmatrix $\boldsymbol{H}$ ergibt sich entsprechend Gleichung (3.31) aus

$$\boldsymbol{H}(\bar{\boldsymbol{s}}(t_k)) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\bar{\boldsymbol{s}}(t_k)} = \left.\left(\frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_1}, \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_1}, \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_2}, \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_2}\right)\right|_{\bar{\boldsymbol{s}}(t_k)}, \tag{3.122}$$

worin für die partiellen Ableitungen folgende Beziehungen gelten:

$$\begin{aligned}
\frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_1} &= -\frac{1}{\|\boldsymbol{X}_{12}\|}\dot{\boldsymbol{X}}_{12} + \frac{\langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{X}_{12}\rangle}{\|\boldsymbol{X}_{12}\|^3}\boldsymbol{X}_{12} \qquad & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_1} &= -\frac{1}{\|\boldsymbol{X}_{12}\|}\boldsymbol{X}_{12} \\
\frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_2} &= \frac{1}{\|\boldsymbol{X}_{12}\|}\dot{\boldsymbol{X}}_{12} - \frac{\langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{X}_{12}\rangle}{\|\boldsymbol{X}_{12}\|^3}\boldsymbol{X}_{12} \qquad & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_2} &= \frac{1}{\|\boldsymbol{X}_{12}\|}\boldsymbol{X}_{12}.
\end{aligned} \tag{3.123}$$

Bei der K-Band-Korrektur hat die Messmatrix $\boldsymbol{H}(\bar{\boldsymbol{s}}(t_k))$ also die Dimension (1×12). Sie ist ebenfalls durch Einsetzen der prädizierten Bahnvektoren bei jedem K-Band-Update erneut zu berechnen.

Die Kovarianzmatrix der Beobachtungen besteht bei den 5-sekündlichen K-Band-Updates nur aus dem konstanten Wert

$$R_k = \sigma_{\dot{\rho}}^2 = \text{const. } \forall\, t_k, \tag{3.124}$$

da keine individuellen Fehlermaße zu jeder K-Band-Messung vorliegen. Für $\sigma_{\dot{\rho}}$ wird deshalb z.B. der in Dunn et al. (2002) genannte Durchschnittswert von $1\,\mu\text{m/s}$ gewählt, der bereits in Tab. 2.1 (Seite 14) erwähnt wurde.

Mit den so ermittelten Größen lässt sich die K-Band-Korrektur nun nach den allgemein gültigen Formeln des Korrekturschritts (Schritt 2 in Tafel 3.2) zu Ende führen und man erhält nach dem ersten K-Band-Update den korrigierten Zustandsvektor $\hat{\boldsymbol{s}}_1$ und dessen Kovarianzmatrix $\widehat{\boldsymbol{P}}_1$ (s. Abb. 3.6). Es folgen dann abwechselnd Prädiktions- und K-Band-Korrekturschritte, bis nach 60 Sekunden neben der K-Band-Messung auch die neuen Navigationslösungen beider Satelliten hinzukommen.

In diesem Fall ist der prädizierte Systemzustand und dessen Kovarianzmatrix auf andere Weise zu verbessern. Die Messgleichung ist nun vektoriell und besteht aus 13 einzelnen Gleichungen:

$$\boldsymbol{b}_k = \begin{pmatrix} \dot{\rho}_{\text{gem}}(t_k) \\ \boldsymbol{X}_{1,\text{nav}}(t_k) \\ \dot{\boldsymbol{X}}_{1,\text{nav}}(t_k) \\ \boldsymbol{X}_{2,\text{nav}}(t_k) \\ \dot{\boldsymbol{X}}_{2,\text{nav}}(t_k) \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{\|\boldsymbol{X}_{12}\|}\langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{X}_{12}\rangle \\ \boldsymbol{X}_1(t_k) \\ \dot{\boldsymbol{X}}_1(t_k) \\ \boldsymbol{X}_2(t_k) \\ \dot{\boldsymbol{X}}_2(t_k) \end{pmatrix}}_{\boldsymbol{h}(\boldsymbol{s})} + \boldsymbol{v}_k. \tag{3.125}$$

Den Vektor der erwarteten Beobachtungen $\boldsymbol{h}(\bar{\boldsymbol{s}}_k)$ erhält man wiederum durch Einsetzen des prädizierten Systemzustands $\bar{\boldsymbol{s}}_k$, d.h. der beiden prädizierten Positions- und Geschwindigkeitsvektoren, in die Funktion $\boldsymbol{h}(\boldsymbol{s})$.

Mit den partiellen Ableitungen von $\boldsymbol{h}$ nach dem Systemzustand

$$\frac{\partial \boldsymbol{h}(\boldsymbol{s})}{\partial \boldsymbol{s}} = \begin{pmatrix} \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_1} & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_2} & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_2} \\ \frac{\partial \boldsymbol{X}_1}{\partial \boldsymbol{X}_1} & \frac{\partial \boldsymbol{X}_1}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \boldsymbol{X}_1}{\partial \boldsymbol{X}_2} & \frac{\partial \boldsymbol{X}_1}{\partial \dot{\boldsymbol{X}}_2} \\ \frac{\partial \dot{\boldsymbol{X}}_1}{\partial \boldsymbol{X}_1} & \frac{\partial \dot{\boldsymbol{X}}_1}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \dot{\boldsymbol{X}}_1}{\partial \boldsymbol{X}_2} & \frac{\partial \dot{\boldsymbol{X}}_1}{\partial \dot{\boldsymbol{X}}_2} \\ \frac{\partial \boldsymbol{X}_2}{\partial \boldsymbol{X}_1} & \frac{\partial \boldsymbol{X}_2}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \boldsymbol{X}_2}{\partial \boldsymbol{X}_2} & \frac{\partial \boldsymbol{X}_2}{\partial \dot{\boldsymbol{X}}_2} \\ \frac{\partial \dot{\boldsymbol{X}}_2}{\partial \boldsymbol{X}_2} & \frac{\partial \dot{\boldsymbol{X}}_2}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \dot{\boldsymbol{X}}_2}{\partial \boldsymbol{X}_2} & \frac{\partial \dot{\boldsymbol{X}}_2}{\partial \dot{\boldsymbol{X}}_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_1} & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_2} & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_2} \\ \multicolumn{2}{c}{\mathbf{I}_{(6\times6)}} & \multicolumn{2}{c}{\mathbf{0}_{(6\times6)}} \\ \multicolumn{2}{c}{\mathbf{0}_{(6\times6)}} & \multicolumn{2}{c}{\mathbf{I}_{(6\times6)}} \end{pmatrix} \tag{3.126}$$

erhält man folgende (13×12)-Messmatrix:

$$\boldsymbol{H}(\bar{\boldsymbol{s}}(t_k)) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\bar{\boldsymbol{s}}(t_k)} = \left.\begin{pmatrix} \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_1} & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_1} & \frac{\partial \dot{\rho}}{\partial \boldsymbol{X}_2} & \frac{\partial \dot{\rho}}{\partial \dot{\boldsymbol{X}}_2} \\ & \mathbf{I}_{(12\times12)} & & \end{pmatrix}\right|_{\bar{\boldsymbol{s}}(t_k)} \tag{3.127}$$

Dabei entspricht die erste Zeile mit den Ableitungen (3.123) der vorher beim K-Band-Korrekturschritt bestimmten $\boldsymbol{H}$-Matrix. Die Kovarianzmatrix des Messrauschens hat jetzt folgende Struktur:

$$\boldsymbol{R}_k = \begin{pmatrix} \sigma^2_{\dot{\rho}} & & & & 0 \\ & \mathrm{cov}(\boldsymbol{X}_{1,\mathrm{nav}},\, \boldsymbol{X}_{1,\mathrm{nav}}) & & & \\ & & \mathrm{cov}(\dot{\boldsymbol{X}}_{1,\mathrm{nav}},\, \dot{\boldsymbol{X}}_{1,\mathrm{nav}}) & & \\ & & & \mathrm{cov}(\boldsymbol{X}_{2,\mathrm{nav}},\, \boldsymbol{X}_{2,\mathrm{nav}}) & \\ 0 & & & & \mathrm{cov}(\dot{\boldsymbol{X}}_{2,\mathrm{nav}},\, \dot{\boldsymbol{X}}_{2,\mathrm{nav}}) \end{pmatrix} \tag{3.128}$$

Neben der Fehlervarianz der K-Band-Messung $\sigma^2_{\dot{\rho}}$ sind darin nur die (3×3)-Kovarianzmatrizen der Positionen bzw. Geschwindigkeiten der beiden Satelliten von Null verschieden, d.h. weitere Korrelationen werden nicht in Betracht gezogen. Die in den Eingangsdaten gegebenen Positionskovarianzen des ersten Satelliten sind in die erste Submatrix nach dem bekannten Schema

$$\mathrm{cov}(\boldsymbol{X}_{1,\mathrm{nav}},\, \boldsymbol{X}_{1,\mathrm{nav}}) = \begin{pmatrix} \sigma^2_{X_{1,\mathrm{nav}}} & \sigma_{X_{1,\mathrm{nav}}Y_{1,\mathrm{nav}}} & \sigma_{X_{1,\mathrm{nav}}Z_{1,\mathrm{nav}}} \\ & \sigma^2_{Y_{1,\mathrm{nav}}} & \sigma_{Y_{1,\mathrm{nav}}Z_{1,\mathrm{nav}}} \\ \text{(sym.)} & & \sigma^2_{Z_{1,\mathrm{nav}}} \end{pmatrix} \tag{3.129}$$

einzusetzen. Die übrigen drei Submatrizen ergeben sich analog.

Damit sind alle Größen spezifiziert, um auch das 60-sekündliche ‚Haupt-Update' nach den allgemeinen Gleichungen des EKF-Korrekturschritts (Tafel 3.2) durchzuführen.

3.2.7 Glättungsmethoden

Anhand des zeitlichen Ablaufs der erweiterten Kalman-Filterung (Abb. 3.5) ist klar geworden, dass bei jedem Messungsupdate durch das Anbringen der geschätzten Korrektur zwangsläufig ein Sprung im Systemzustand entsteht. Deshalb weisen auch die in dieser Arbeit verdichteten GRACE-Orbits zwischen den betreffenden Bahnbogenstücken Sprünge auf, also alle 5 bzw. 60 Sekunden. Derartige bei einer Nachprozessierung entstehende, regelmäßige Unstetigkeiten sind, zumindest aus physikalischer Sicht, nicht realistisch und aus diesem Grund nach Möglichkeit zu vermeiden.

In der Literatur zur Kalman-Filterung ist häufig von Glättungsfiltern zu lesen, welche die Daten wie bei der Kalman-Filterung von ‚vorne nach hinten' und anschließend nochmals ‚von hinten nach vorne' abarbeiten (s. z.B. „forward-backward smoothing" in Simon, 2006, S. 280). In dieser Anwendung würde sich auch ein erweitertes Glättungsfilter nach Rauch, Tung und Striebel[1] anbieten.

Da die Glättungsfilter für sich genommen schon ein weites Themenfeld bilden, die Umsetzung derselben erneut intensive Einarbeitung erfordern würde, und auch nicht sicher ist, in welchem Ausmaß sich dadurch die Sprünge im Vergleich zur Kalman-Filterung reduzieren, werden in dieser Arbeit später nur drei einfache Möglichkeiten zur nachträglichen Glättung der verdichteten Orbits untersucht: die kubische Spline-Interpolation, das Einfach-Schießverfahren und die kleinste-Quadrate-Einpassung (s. Abschnitt 5.3.6).

Die weiterführende Möglichkeit, die Orbitverdichtung mittels eines Glättungsfilters umzusetzen, sei daher für zukünftige Arbeiten vorgemerkt (s.a. Schlussbem. auf S. 133).

1 Herbert E. Rauch, F. Tung und Charlotte T. Striebel veröffentlichten ihre Glättungsmethode, auf dem auch dieses erweiterte Glättungsfilter basiert, erstmals 1965 für ein lineares dynamisches System, siehe Rauch et al. (1965).

4

Kapitel 4

Implementierung und numerische Untersuchungen

In diesem Kapitel wird über einige Teile der in MATLAB realisierten Toolbox Auskunft gegeben und anschließend werden verschiedene numerische Untersuchungen anhand der implementierten Funktionen durchgeführt.

Da die Quelltexte zahlreich und mitunter sehr umfangreich sind, kann hier verständlicherweise nicht jede Codezeile erklärt werden. Grundsätzlich ist die Toolbox modular aufgebaut mit Hauptprogrammen sowie Haupt- und Unterfunktionen, die jeweils nur einen Teilaspekt zur Gesamtlösung beitragen. Die Aufgaben der Hauptprogramme und einiger relevanter Funktionen sind in Anhang A ab Seite 135 beschrieben. Eine Übersicht und kurze Beschreibung aller enthaltenen Dateien ist in der Textdatei `contents.txt` zu finden.

Nachfolgend wird zuerst der Aufbau des Hauptprogramms `ekf_orbit_main` zur Prozessierung realer GRACE-Daten sowie die Struktur ebendieser Eingangsdaten näher erläutert.

4.1 Hauptprogramm zur Verdichtung von GRACE-Satellitenorbits

4.1.1 Einleseteil und Struktur der Eingangsdaten

Das Hauptprogramm `ekf_orbit_main` ist gegliedert in einen Einlese-, Verarbeitungs- und Ausgabeteil. Im Einleseteil erwartet das Programm drei Eingangsdateien im `mat`-Dateiformat, deren Pfade nach dem Ausführen erfragt werden:

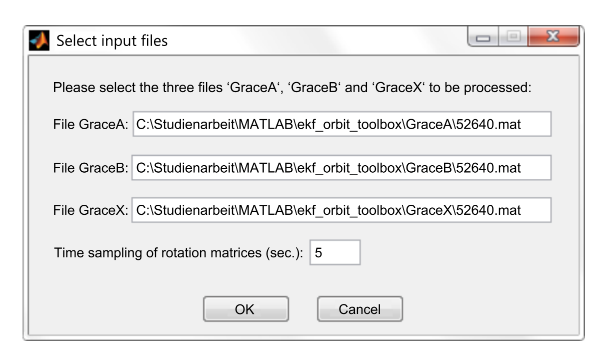


Abb. 4.1: Programmfenster zur Auswahl der Eingangsdateien

Der Toolbox liegen drei Beispieldateien in den Unterverzeichnissen `\GraceA`, `\GraceB` und `\GraceX` bei, die u.a. die realen GRACE-Daten (Navigationslösungen und K-Band-Beobachtungen) vom 1. Jan. 2003 (52640 MJD) beinhalten (s. auch Abschn. 5.1). In der ersten, standardmäßig im Unterverzeichnis `\GraceA` gesuchten Datei müssen die Variablen

`navpos`, `navvel`, `navpos_i`, `navvel_i`, und `Ri2e`

vorhanden sein. Bei `navpos` und `navvel` handelt es sich um die GNV1B-Navigationslösung (des GRACE-A-Satelliten) bezüglich des geozentrisch gelagerten, erdfesten Systems. Diese Daten werden

im Hauptprogramm `ekf_orbit_main` nicht zur eigentlichen Verdichtung der Satellitenbahn, sondern erst im Ausgabeteil zur Darstellung der Ergebnisse auf einer Plattkarte genutzt. Verarbeitet wird stattdessen die im geozentrisch-raumfesten Bezugssystem gegebene Navigationslösung. Die raumfesten Satellitenpositionen liegen in `navpos_i` als Zeitreihe mit 60-Sekunden-Sampling in Matrixform vor, wobei jeder Zeitpunkt einer Zeile entspricht und zu jedem Zeitpunkt elf Größen in den Spalten abgespeichert sind:

$$\begin{bmatrix} \overset{1}{\text{MJD}} & \overset{2}{t} & \overset{3}{X_{\text{nav}}} & \overset{4}{Y_{\text{nav}}} & \overset{5}{Z_{\text{nav}}} & \overset{6}{\sqrt{\sigma_{XX}}} & \overset{7}{\sqrt{\sigma_{XY}}} & \overset{8}{\sqrt{\sigma_{XZ}}} & \overset{9}{\sqrt{\sigma_{YY}}} & \overset{10}{\sqrt{\sigma_{YZ}}} & \overset{11}{\sqrt{\sigma_{ZZ}}} \end{bmatrix}$$

Die erste Spalte ist das in Tagen gegebene modifizierte julianische Datum (MJD). Zur Umrechnung desselben in einen Zeitvektor bezüglich des Gregorianischen Kalenders dient in der Toolbox übrigens die Funktion `mjd2greg`. Die zweite Zeitvariable t stellt die Zeit in Sekunden seit Beginn dieses Tages (00:00 Uhr GPS-Zeit) dar und deckt dementsprechend den Zeitbereich von 0 bis 86340 s ab. Die übrigen Spalten von `navpos_i` haben die Einheit Meter. Diese Matrixstruktur gilt analog für die Variablen `navvel_i`, `navpos` und `navvel`. Bei den Geschwindigkeiten (`navvel(_i)`) ist entsprechend die Einheit der dritten bis elften Spalte m/s.

Die Variable `Ri2e` enthält die zugehörigen Transformationsmatrizen vom raumfesten ins erdfeste Bezugssystem. Diese liegen als Zeitreihe wiederum in einem Zeilenformat vor, wobei die neun Spalten wie folgt zu den Einträgen der Rotationsmatrix korrespondieren:

$$\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}(t) = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad \longleftrightarrow \quad \begin{bmatrix} \overset{1}{R_{11}} & \overset{2}{R_{12}} & \overset{3}{R_{13}} & \overset{4}{R_{21}} & \dots & \overset{9}{R_{33}} \end{bmatrix}$$

Da in der Matrix `Ri2e` ein expliziter Zeitbezug fehlt, wird davon ausgegangen, dass die erste Zeile dem ersten Beobachtungszeitpunkt der Navigationslösung entspricht. Zudem muss aus diesem Grund auch das Sampling der Rotationsmatrizen angegeben werden. Bei der bereitgestellten Zeitreihe beträgt dieses 5 Sekunden. Jede in Zeilenform gespeicherte Rotationsmatrix in `Ri2e` kann man leicht mittels den MATLAB-Funktionen `vec2mat` oder `reshape` in die zugehörige Matrixdarstellung überführen; für den umgekehrten Weg dient die selbst erstellte Funktion `mat2vec`.

Die Eingangsgrößen, die für den ersten GRACE-Satelliten in der ersten `mat`-Datei vorliegen, sind auch entsprechend für den zweiten Satelliten in einer `mat`-Datei vorzugeben (Dateipfad `GraceB`). Die dritte Eingangsdatei (`GraceX`-Pfad) beinhaltet in der Matrix `KBR1B_X` die K-Band-Beobachtungen als Zeitreihe im 5-Sekunden-Sampling:

$$\begin{bmatrix} \overset{1}{\text{MJD (d)}} & \overset{2}{t\ \text{(s)}} & \overset{3}{\rho\ \text{(m)}} & \overset{4}{\dot{\rho}\ (\text{m/s})} & \overset{5}{\ddot{\rho}\ (\text{m/s}^2)} \end{bmatrix}$$

Hierin weist die Relativentfernung ρ wie schon erwähnt aufgrund der ungelösten Phasenmehrdeutigkeit einen unbekannten Offset auf.

Beim Einlesevorgang werden die Variablen neu benannt, sodass nun nachstehende Variablen im Arbeitsbereich zur Verfügung stehen:

`Ri2eA`, `Ri2eB`	Rotationsmatrizen für GRACE-A und -B
`kband`	K-Band-Beobachtungszeitreihe
`navposA_i`, `navvelA_i`	Navigationslösung für GRACE-A mit raumfestem Bezug
`navposB_i`, `navvelB_i`	Navigationslösung für GRACE-B mit raumfestem Bezug
`navposA_e`, `navvelA_e`	Navigationslösung für GRACE-A mit erdfestem Bezug
`navposB_e`, `navvelB_e`	Navigationslösung für GRACE-B mit erdfestem Bezug

Anschließend können in einem Programmfenster (GUI) wichtige Optionen für die weitere Verarbeitung der Daten festgelegt werden (s. Abb. 4.1). Zuerst kann der Nutzer bestimmen, bis zu welchem

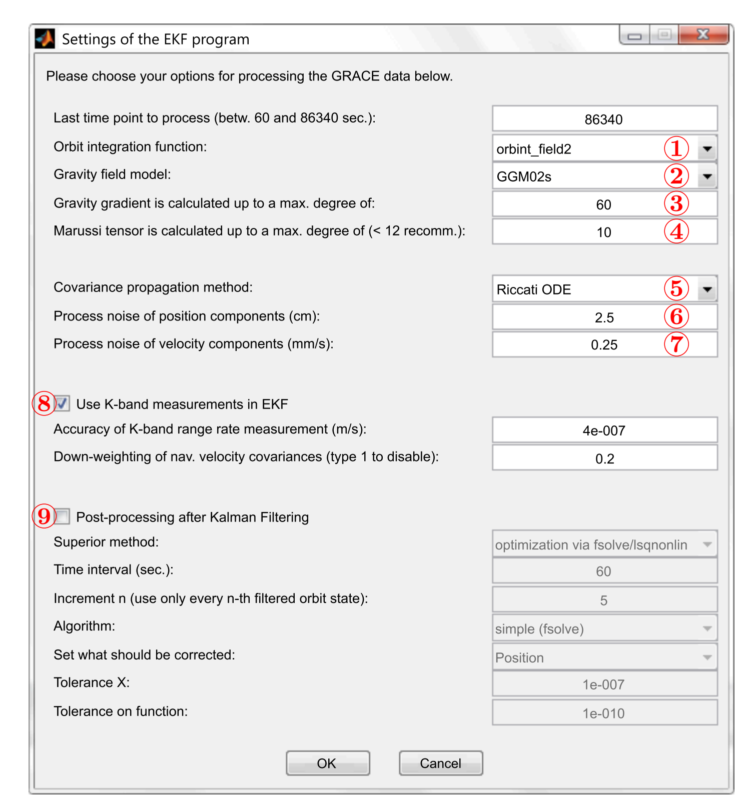


Abb. 4.2: Programmfenster mit den Optionen für die Verarbeitung der Daten

Zeitpunkt die Daten prozessiert werden sollen. Ferner kann er das bei der Orbitintegration genutzte Gravitationsfeld ② wählen, welches im Anschluss aus dem Unterverzeichnis `\gravfields` eingelesen und dann in die Variable `field` geschrieben wird. Je nach ausgewähltem Gravitationsfeldmodell werden die mit dem Modell verknüpften Konstanten[1] aus der entsprechenden Klassendefinition im Verzeichnis `\gravfields` geladen und in der Variable `cstruct` abgespeichert. `cstruct` wird dann an die Funktionen `ekf_orbit(2)` übergeben, sodass alle Unterfunktionen bei Bedarf auf die Konstanten zugreifen können.

Im ‚Drop-Down'-Menü ① lässt sich der Orbitintegrator auswählen, wodurch intern der Parameter `orbint_sel` festgelegt wird. Wählt man hier die Option `orbint_field + 'PetVer'`, so erfolgt die Berechnung des Marussi-Tensors nach Petrovskaya und Vershkov mittels der Funktion `tenspshs`. Das Eingabefeld ③ dient dazu, den maximalen Entwicklungsgrad bei der Bestimmung des Gravitationsgradienten `lmax` festzulegen (nur relevant bei den Integratoren `orbint_field` und `orbint_field2`). Im Feld ④ kann man zudem den Entwicklungsgrad für den Marussi-Tensor angeben; die dabei angelegte Variable `lmax2` wird nur benötigt, falls der Integrator `orbint_field` ausgewählt ist. Unter Punkt ⑤ ist eine Variante aus den drei in Abschnitt 3.2.4.1 eingeführten Methoden zur Prädiktion der Schätzfehlerkovarianzmatrix zu selektieren, wodurch der Parameter `cov_sel` gesetzt wird. Durch die beiden Eingaben ⑥ und ⑦ werden die Parameter `qpos` und `qvel` erfragt, welche für die jeweils angenommene Matrix des Systemrauschens maßgeblich sind[2]. Bestätigt man die ‚Checkbox' ⑧, erfolgt die Orbitverdichtung mittels der Funktion `ekf_orbit2` – ansonsten mit der Funktion `ekf_orbit` – und zwei weitere Parameter sind mit Werten zu belegen: erstens die Standardabweichung der beobachteten K-Band-Relativgeschwindigkeiten, welche bei jeder K-Band-Korrektur berücksichtigt wird. Dies ist nötig, weil im KBR1B-Datensatz keine Fehlermaße zu den einzelnen Messungen verzeichnet sind. Zweitens ist der Parameter $\sigma_{0,\texttt{navvel}}$ festzulegen, der dazu dient, die in `navvel_i` vorliegenden Fehlermaße der GPS-Geschwindigkeiten herunterzugewichten. Dies wird erreicht, indem man in der Funktion `ekf_orbit2` beim 60-Sek.-Korrekturschritt die (3×3)-Submatrizen $\mathrm{cov}(\dot{\boldsymbol{X}}_{1,\mathrm{nav}}, \dot{\boldsymbol{X}}_{1,\mathrm{nav}})$ und $\mathrm{cov}(\dot{\boldsymbol{X}}_{2,\mathrm{nav}}, \dot{\boldsymbol{X}}_{2,\mathrm{nav}})$ der Matrix

1 Dabei handelt es sich um folg. Konstanten: geozentrische Gravitationskonstante (`GM`), große Halbachse des Ellipsoids (`ae`) und unnormierter „zero-tide"-J_2-Koeffizient (`J2`).

2 Wie man hier en détail mit diesen Parametern weiterverfährt, wird sogleich in Abschnitt 4.4 dargelegt.

$\boldsymbol{R}_k$ (s. Gl. (3.128)) durch $\sigma^2_{0,\texttt{navvel}}$ dividiert. In Abschnitt 4.7 und 5.3 wird noch deutlich, warum die beiden Parameter unter Punkt ⑧ beim ‚Fein-Tuning' des Filters wichtig sein können.

Die letzte Option ⑨ regelt die Nachprozessierung des verdichteten Orbits mit einfachen Möglichkeiten zur Glättung und ist eher experimentell zu sehen (Näheres hierzu später in Abschn. 5.3.6).

4.1.2 Verarbeitungs- und Ausgabeteil

Je nach gewählter Option ⑧ verdichtet man die eingelesenen Navigationslösungen beider Satelliten im Verarbeitungsteil des Hauptprogramms durch zweimaliges Aufrufen der Funktion `ekf_orbit` oder – bei Einbeziehung der K-Band-Relativgeschwindigkeiten – durch einmaliges Aufrufen der Funktion `ekf_orbit2`, wobei man jeweils auch die vom Nutzer definierten Optionen (Abb. 4.2) an die betreffende Funktion übergibt. Auf diese beiden Hauptfunktionen wird im nachstehenden Abschnitt nochmals kurz eingegangen. Sie liefern nun im Hauptprogramm die verdichteten Bahndaten für beide Satelliten in einer zeitlichen Diskretisierung von einer Sekunde[1]. Diese Daten sind in den Variablen

`intposA_i`, `intvelA_i`, `intposB_i`, `intvelB_i`

im Arbeitsbereich vorzufinden und können sogleich zusammen mit den Einstellungen (Variable `dlg_answer_ekf`) im Verzeichnis `\ekf_orbit_out` abgespeichert werden[2]. Die Datenstruktur der vier Ergebnisvariablen (`intpos_i` usw.) entspricht dem oben angegebenen Datenformat der Navigationslösung (`navpos_i` usw.).

Schließlich wird noch ein statistischer Vergleich zwischen den verdichteten Orbits und den K-Band-Beobachtungen im „command window" ausgegeben. Außerdem kann man sich mittels eines Dialogfensters verschiedene Ergebnisgrafiken anzeigen lassen und diese auf Wunsch sichern.

4.2 Hauptfunktionen zur EKF-Orbitverdichtung

Es wurde bereits angemerkt (in Abschn. 3.2.6, S. 57), dass in dieser Arbeit zwei Filtervarianten bzw. Hauptfunktionen zur Orbitverdichtung mithilfe der erweiterten Kalman-Filterung erstellt wurden. Während die erste Funktion `ekf_orbit` nur einzelne Satellitenbahnen verdichtet, ist die zweite, für GRACE entworfene Funktion `ekf_orbit2` imstande, zwei GRACE-Orbits inklusive der K-Band-Messungen $\dot{\rho}$ zu verarbeiten und dementsprechend gleich beide verdichtete Bahnen zu liefern.

In beiden Funktionen werden nach einer Initialisierung (s. nächster Abschnitt) die Zeilen der gegebenen Beobachtungszeitreihen durch Schleifen im typischen Regelkreis von Prädiktion und Korrektur abgearbeitet. Für den Prädiktionsschritt nutzen beide Funktionen die Unterfunktion `ekf_predict`, welche Gegenstand des übernächsten Abschnitts ist. Der prinzipielle Berechnungsablauf der Hauptfunktionen ist anhand der ausreichend kommentierten Quellcodes nachvollziehbar. Nachfolgend wird daher nur die unterschiedliche Vorgehensweise hinsichtlich des Korrekturschritts beschrieben.

Hauptfunktion `ekf_orbit`

Der Korrekturschritt in `ekf_orbit` beginnt damit, indem der jeweils neue Messvektor und dessen Kovarianzmatrix aus den Zeitreihen der gegebenen Daten des zu verdichtenden Orbits extrahiert werden. Diese Variablen werden anschließend zusammen mit dem zuvor bis zu diesem Messzeitpunkt

1 Die Diskretisierung von einer Sekunde ist in den Hauptfunktionen fest definiert.

2 Dabei wird im Verzeichnis `\ekf_orbit_out` noch ein Unterordner angelegt, dessen Name sich aus dem MJD und der Systemzeit nach Abschluss der Prozessierung zusammensetzt.

prädizierten Systemzustand und der zugehörigen Schätzfehlerkovarianzmatrix sowie mit der erwarteten Beobachtung $\boldsymbol{h}(\bar{\boldsymbol{s}}_k)$ und der Messmatrix $\boldsymbol{H}(\bar{\boldsymbol{s}}_k)$ an die Funktion `ekf_update` (s. Anhang A.2, S. 138) übergeben. Ebendiese Funktion ist in der Toolbox dafür zuständig, die eigentliche Korrektur nach den allgemeinen Filtergleichungen (Schritt 2 in Tafel 3.2) zu bewerkstelligen. Entsprechend Abschnitt 3.2.6.1 wird dabei in `ekf_orbit` der Vektor $\boldsymbol{h}(\bar{\boldsymbol{s}}_k)$ mit dem prädizierten Bahnvektor und die Messmatrix mit der (6×6)-Einheitsmatrix gleichgesetzt.

Falls in den gegebenen Orbits Datenlücken vorkommen, die mit `NaN` gekennzeichnet sind, werden sie bei der Verarbeitung übersprungen und es wird mit dem Bahnabschnitt fortgefahren, dessen Anfangs- und Endvektor keine `NaN`-Einträge mehr aufweisen. Allerdings handelt es sich bei den hier genutzten GNV1B-Orbits nicht etwa um rein kinematische, sondern um vorverarbeitete, reduziert-dynamische Orbits, die keine Lücken (mehr) haben.

Hauptfunktion `ekf_orbit2`

In `ekf_orbit2` integriert man die zwei Bahnen separat (jeweils mit `ekf_predict`) bis zum Messungsupdate, erweitert dann den Zustandsvektor gemäß Gl. (3.118) (Abschn. 3.2.6.2, S. 59) zu

$$\bar{\boldsymbol{s}}_k = (\bar{\boldsymbol{X}}_1,\ \dot{\bar{\boldsymbol{X}}}_1,\ \bar{\boldsymbol{X}}_2,\ \dot{\bar{\boldsymbol{X}}}_2)^\top$$

und setzt auch die beiden prädizierten (6×6)-Kovarianzmatrizen $\bar{\boldsymbol{P}}_k$ zu einer (12×12)-Kovarianzmatrix zusammen. Bevor der Algorithmus `ekf_update` wiederum die eigentliche Aufdatierung erledigt, wird die Fallunterscheidung getroffen, ob am aktuellen Zeitpunkt ein K-Band- oder ein Haupt-Update durchzuführen ist und dementsprechend werden die restlichen benötigten Variablen, d.h. der Beobachtungsvektor $\boldsymbol{b}_k$, dessen Kovarianzmatrix $\boldsymbol{R}_k$, die erwartete Beobachtung $\boldsymbol{h}(\bar{\boldsymbol{s}}_k)$ und die Messmatrix $\boldsymbol{H}(\bar{\boldsymbol{s}}_k)$, gemäß den Formeln in Abschnitt 3.2.6.2 aufgestellt.

Dabei wird die erwartete K-Band-Beobachtung $\dot{\rho}$ im Vektor $\boldsymbol{h}(\bar{\boldsymbol{s}}_k)$ mithilfe der Funktion `grace_parms` (s. Anhang A.2, S. 138) aus den prädizierten Zustandsvektoren beider Satelliten erhalten.

Die Messmatrix $\boldsymbol{H}$ berechnet man im Fall der K-Band-Updates in `H_kband` und im Fall der Haupt-Updates in `H_nav`. Beides sind eingebettete Unterfunktionen der Funktion `ekf_orbit2`. Anstatt nach den Formeln (3.123) drei Einträge der Messmatrix gemeinsam zu ermitteln, werden in diesen Unterfunktionen, da es sich hier als numerisch vorteilhaft erwiesen hat, die Ableitungen von $\dot{\rho}$ separat nach jeder Komponenten des Systemzustands berechnet, um die zwölf K-Band-Komponenten der Messmatrix (jeweils erste Zeile) aus diesen skalaren Werten zusammenzusetzen. Analytisch gesehen ist dies aber äquivalent zu den vektoriellen Gleichungen.

Nach erfolgter Korrektur wird $\hat{\boldsymbol{s}}_k$ und $\widehat{\boldsymbol{P}}_k$ ‚zerlegt', um die Orbits wieder separat zu prädizieren.

Die verwendeten K-Band-Daten sind zwar von Ausreißern befreit, enthalten jedoch im Gegensatz zu den Navigationslösungen hin und wieder Datenlücken, die durch `NaN`-Einträge in den Zeilen des Datensatzes kenntlich gemacht sind. Bei der Verarbeitung mit der Funktion `ekf_orbit2` hat man sich ebenfalls für die ‚radikale' Methode entschieden, die betreffenden 60-Sekunden-Bahnabschnitte zu überspringen und mit `NaN`-Einträgen auszufüllen. Dies wird ggf. durch entsprechende Warnmeldungen im „command window" angekündigt.

4.3 Filterinitialisierung

Die Initialisierung des Kalman-Filters betrifft die Frage, wie die Anfangswerte für die erste Prädiktion zu wählen sind. Bisher wurde davon ausgegangen, dass der anfängliche Systemzustand $\hat{\boldsymbol{s}}_0$ und die Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ bereits vorliegen.

Generell müssen die Anfangswerte $\hat{\boldsymbol{s}}_0$ insofern realistisch sein, dass die damit integrierte Bahn auch einigermaßen zu den späteren Messungen passt. Falls die genutzten Startwerte für den Systemzustand als unsicher betrachtet werden, sollten die Einträge der zugeordneten Kovarianzmatrix ausreichend groß sein, damit bei den ersten Messungen eine angemessene Korrektur erfolgen kann. Darüberhinaus sollte man die Symmetrie und positive Definitheit von $\widehat{\boldsymbol{P}}_0$ sicherstellen, denn nicht positiv-definite Kofaktormatrizen führen meist zur Filterdivergenz.

Häufig, vor allem im Falle einer (quasi-)linearen Messfunktion $\boldsymbol{h}$, können die Startwerte (näherungsweise) aus der ersten Beobachtung $\boldsymbol{b}_0$ mit bekanntem Messrauschen $\boldsymbol{R}_0$ durch die Beziehungen

$$\hat{\boldsymbol{s}}_0 \approx (\boldsymbol{H}_0)^{-1}\boldsymbol{b}_0 \tag{4.1}$$

$$\widehat{\boldsymbol{P}}_0 \approx (\boldsymbol{H}_0)^{-1}\boldsymbol{R}_0(\boldsymbol{H}_0^\top)^{-1} \tag{4.2}$$

abgeschätzt werden. Hier wird das jeweilige Filter folgendermaßen initialisiert:

- Im Fall der Funktion `ekf_orbit` wird einfach der erste beobachtete Zustandsvektor (d.h. der erste Zustandsvektor des gegebenen Orbits) als Ausgangszustand verwendet:

$$\hat{\boldsymbol{s}}_0 := \boldsymbol{b}_0 = \begin{pmatrix} \boldsymbol{X}_{\text{nav}}(t_0) \\ \dot{\boldsymbol{X}}_{\text{nav}}(t_0) \end{pmatrix}$$

  Dementsprechend wird die Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ analog zu Gl. (3.117) aus den bereitgestellten zwölf Messunsicherheitswerten der ersten Beobachtung zusammengesetzt. Diese Gleichsetzung der Anfangswerte mit den entsprechenden Messwerten ist natürlich nur deshalb so einfach möglich, da wir mit Gleichung (3.114) implizit angenommen haben, dass man den Systemzustand direkt im Zustandsraum beobachtet, bzw. für die Messmatrix $\boldsymbol{H} = \mathbf{I}$ gilt.

- Im Fall der Funktion `ekf_orbit2` ist die Beobachtungsfunktion $\boldsymbol{h}$ in der K-Band-Komponente nicht-linear mit dem Systemzustand verknüpft. Daher werden die Anfangswerte in der Unterfunktion `ini_update` durch eine ‚Vorab-K-Band-Korrektur' bestimmt. Dabei wird aus den ersten Navigationslösungen der beiden Satelliten der 12-dimensionale Systemzustand $\bar{\boldsymbol{s}}_0$ und die zugehörige Kovarianzmatrix $\overline{\boldsymbol{P}}_0$ gebildet; mit der K-Band-Relativgeschwindigkeit als Beobachtung werden dann völlig analog zum 5-sekündlichen K-Band-Update mithilfe der Funktion `ekf_update` die korrigierten Anfangswerte $\hat{\boldsymbol{s}}_0$ und $\widehat{\boldsymbol{P}}_0$ berechnet.

Zugegebenermaßen ist es im Allgemeinen durchaus riskant, lediglich die Beobachtungen des Startzeitpunkts zur Filterinitialisierung zu nutzen, denn falls die erste Beobachtung z.B. zufällig ein ‚Ausreißer' sein sollte, könnte das Filter möglicherweise divergieren. Diese Gefahr besteht bei den GNV1B-Daten kaum, weil es sich dabei um reduziert-dynamische Orbits handelt (vgl. Abschn. 5.1).

Alternativ wäre für die Initialisierung nicht-vorprozessierter Orbits z.B. denkbar, ausgehend vom ersten gemessenen Bahnpunkt eine Bahn zum zweiten zu integrieren und mittels des Schießverfahrens die Anfangswerte zu verbessern oder die Bahn noch weiter zu integrieren und die Differenz zu mehreren gemessenen Bahnpunkten durch kleinste-Quadrate-Schätzung iterativ zu minimieren (s. Funktion `shootorb`, Anhang A.2, S. 142), um auf diese Weise verbesserte Anfangswerte zu erhalten, die dann mit geringerem Risiko als Startwerte für das Filter benutzt werden können.

4.4 Implementierung der Orbitintegration

Zur Prädiktion eines korrigierten Ausgangszustands und dessen Kovarianzmatrix von einem Messzeitpunkt zum nächsten dient in der realisierten Toolbox die Funktion `ekf_predict`, die unter Berücksichtigung der im Programmfenster (Abb. 4.2) gewählten Optionen den Orbitintegrator

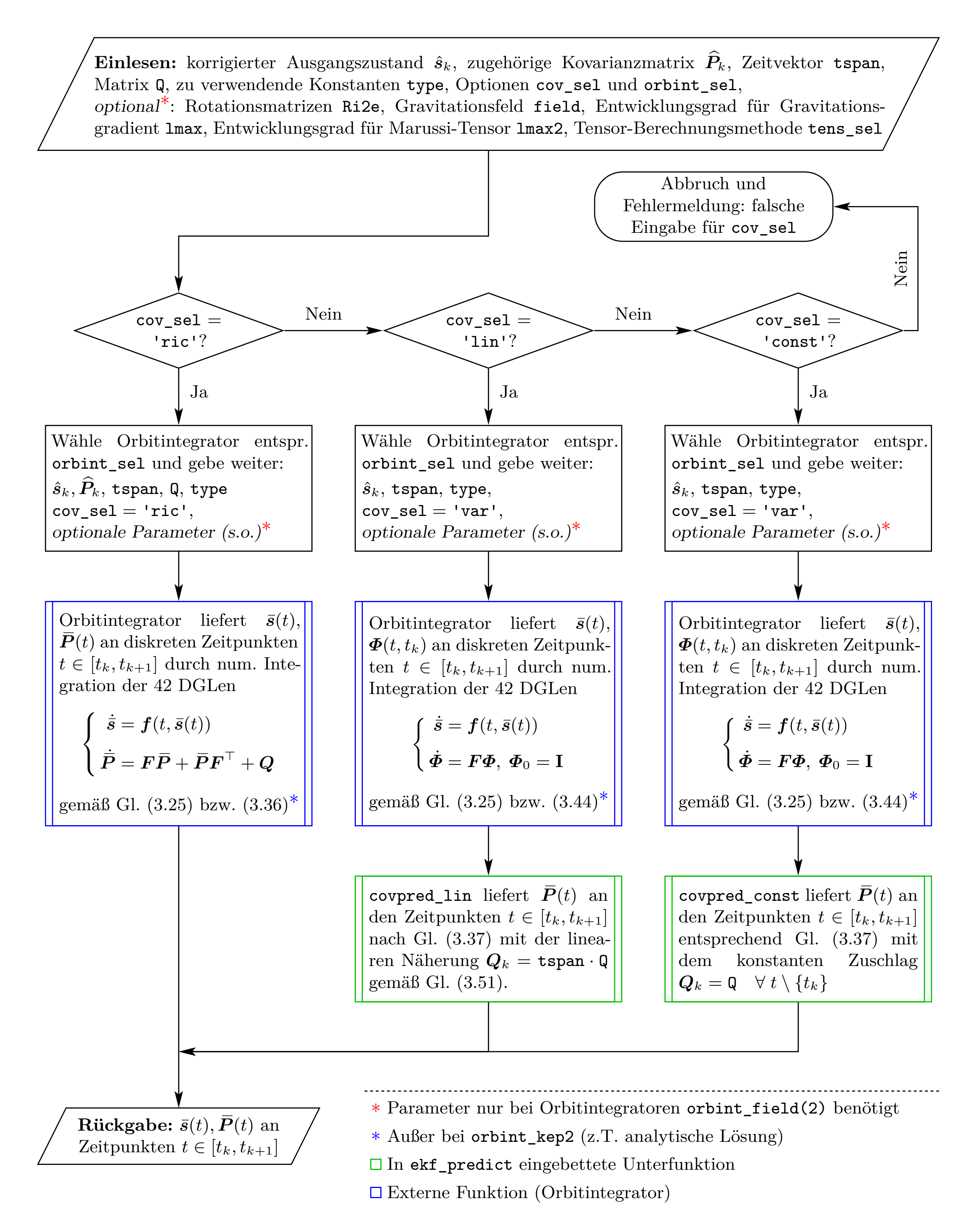


Abb. 4.3: Ablaufdiagramm der Funktion ekf_predict, die zur Prädiktion der Systemzustände und Schätzfehlerkovarianzmatrizen dient.

mit den entsprechenden Einstellungen aufruft und den propagierten Systemzustand und dessen Kovarianzmatrix zurückliefert.

Entsprechend den Festlegungen in Abschnitt 3.2.4.1 kann zur Prädiktion der Schätzfehlerkovarianzmatrizen zwischen den drei Methoden `'ric'`, `'lin'` und `'const'` gewählt werden und dementsprechend ist ein bestimmtes DGL-System simultan zu integrieren: in den Fällen `'const'` und `'lin'` ist die nicht-lineare Systemgleichung zusammen mit der Variationsgleichung

$$\begin{cases} \dot{\bar{\boldsymbol{s}}}(t) &= \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) = \left(\dot{\boldsymbol{X}}(t),\ \boldsymbol{f}_{\mathrm{K}}(t, \boldsymbol{X}(t), \boldsymbol{p})\right)^\top \quad , \quad \bar{\boldsymbol{s}}(t_k) = \hat{\boldsymbol{s}}_k \\ \dot{\boldsymbol{\Phi}}(t, t_k) &= \boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})\, \boldsymbol{\Phi}(t, t_k) \quad , \quad \boldsymbol{\Phi}(t_k, t_k) = \mathbf{I} \end{cases}$$

numerisch zu integrieren und im Fall `'ric'` ist die Matrix-Riccati-DGL simultan mit der Systemgleichung

$$\begin{cases} \dot{\bar{\boldsymbol{s}}}(t) &= \boldsymbol{f}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) = \left(\dot{\boldsymbol{X}}(t),\ \boldsymbol{f}_{\mathrm{K}}(t, \boldsymbol{X}(t), \boldsymbol{p})\right)^\top \quad , \quad \bar{\boldsymbol{s}}(t_k) = \hat{\boldsymbol{s}}_k \\ \dot{\overline{\boldsymbol{P}}}(t) &= \boldsymbol{F}(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p})\ \overline{\boldsymbol{P}}(t) + \overline{\boldsymbol{P}}(t)\ \boldsymbol{F}^\top(t, \bar{\boldsymbol{s}}(t), \boldsymbol{p}) + \boldsymbol{Q}(t) \quad , \quad \overline{\boldsymbol{P}}(t_k) = \widehat{\boldsymbol{P}}_k \end{cases}$$

numerisch zu lösen. Formuliert man dies jeweils in ein vektorielles DGL-System um, liegen in beiden Fällen insgesamt 42 DGLen vor, da zu den sechs DGLen der Systemgleichung die 36 einzelnen Gleichungen des (6×6)-Matrix-AWPs hinzukommen.

Das Ablaufdiagramm der Funktion `ekf_predict` in Abb. 4.3 zeigt, wie die drei Methoden `'ric'`, `'lin'` und `'const'` zusammen mit den Orbitintegratoren realisiert wurden. Um eine der drei Methoden anzuwenden, ist die Eingangsvariable `cov_sel` entsprechend zu belegen (mit `'ric'` oder `'lin'` oder `'const'`). Mit dem Eingangsargument `orbint_sel` wird einer der fünf implementierten Orbitintegratoren ausgewählt (siehe Tabelle 4.1).

Tab. 4.1: Übersicht der implementierten Orbitintegratoren

Integrator	`orbint_sel`	Typ	Kräftemodell
`orbint_field`	`'field'`	numerisch	vollständiges Gravitationsfeldmodell für Kräftefunktion und Marussi-Tensor (s. Abschn. 3.2.5.1)
`orbint_field2`	`'field2'`	numerisch	vollst. Gravitationsfeldmodell für Kräftefunktion, Marussi-Tensor bis Grad 2 (Abschn. 3.2.5.1 u. 3.2.5.2)
`orbint_j2`	`'j2'`	numerisch	Gravitationsfeldmodell bis Grad 2 für Kräftefunktion und Marussi-Tensor (s. Abschn. 3.2.5.2)
`orbint_kep`	`'kep'`	numerisch	ungestörte Keplerbewegung (Abschnitt 3.2.5.3)
`orbint_kep2`	`'kep2'`	größtenteils analytisch	ungestörte Keplerbewegung (Abschnitt 3.2.5.4)

In den Fällen `'lin'` und `'const'` für `cov_sel` gibt `ekf_predict` an den gewählten Orbitintegrator nur `cov_sel = 'var'` weiter, weil in beiden Fällen der Orbitintegrator dasselbe DGL-System lösen muss. Daher akzeptieren die Orbitintegratoren für die Eingangsvariable `cov_sel` nur die zwei Möglichkeiten `'ric'` und `'var'` und liefern somit neben den prädizierten Bahnvektoren entweder die zugehörigen Kovarianzmatrizen $\overline{\boldsymbol{P}}(t)$ oder die Transitionsmatrizen $\boldsymbol{\Phi}(t, t_k)$ zurück. Wie im Flussdiagramm dargestellt, werden in den Fällen `'lin'` und `'const'` die prädizierten Schätzfehlerkovarianzmatrizen im Anschluss nach Gleichung (3.37) gemäß

$$\overline{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k \widehat{\boldsymbol{P}}_k \boldsymbol{\Phi}_k^\top + \boldsymbol{Q}_k$$

aus der geschätzten Kovarianzmatrix am Ausgangszeitpunkt, den Transitionsmatrizen und der Matrix `Q` in Unterfunktionen von `ekf_predict` berechnet. `Q` ist eine spezielle Eingangsgröße, die zur Berechnung der Spektraldichte- oder Kovarianzmatrix des Systemrauschens dient und bereits vorab in allen Hauptprogrammen folgendermaßen festgelegt wird:

Im Fall der Orbitverdichtung mittels der Funktion `ekf_orbit` (60-sekündiges Integrationsintervall) und der Methode `'const'` für das Systemrauschen stellt man die Matrix `Q` nach der Beziehung

$$\texttt{Q} = \begin{pmatrix} \texttt{qpos}^2 \cdot \mathbf{I}_{(3\times3)} & \mathbf{0}_{(3\times3)} \\ \mathbf{0}_{(3\times3)} & \texttt{qvel}^2 \cdot \mathbf{I}_{(3\times3)} \end{pmatrix} \tag{4.3}$$

auf, mit individuell in jeder Verarbeitung festzulegenden Werten `qpos` und `qvel`. In diesem Fall stellt `Q` also die konstante Kovarianzmatrix des Systemrauschens dar und wird in dieser Form auf alle prädizierten Schätzfehlerkovarianzmatrizen $\boldsymbol{\Phi}_k \widehat{\boldsymbol{P}}_k \boldsymbol{\Phi}_k^\top$ addiert.

Sind die K-Band-Beobachtungen in die Orbitverdichtung miteinzubeziehen (Funktion `ekf_orbit2`), erfolgt alle 5 Sekunden die K-Band-Korrektur. Falls außerdem die Methode `'const'` gewählt ist, wird obige Matrix durch die Anzahl der Prädiktionsschritte bis zum ‚Haupt-Update' nach 60 Sekunden geteilt:

$$\texttt{Q} = \frac{1}{\Delta t_{\text{nav}}/\Delta t_{\text{kbr}}} \begin{pmatrix} \texttt{qpos}^2 \cdot \mathbf{I}_{(3\times3)} & \mathbf{0}_{(3\times3)} \\ \mathbf{0}_{(3\times3)} & \texttt{qvel}^2 \cdot \mathbf{I}_{(3\times3)} \end{pmatrix} \tag{4.4}$$

Die Zeitabstände gewinnt man dabei aus den Datensätzen (hier: $\Delta t_{\text{nav}} = 60\,\text{s}$, $\Delta t_{\text{kbr}} = 5\,\text{s}$). Folgende Überlegungen führten zu dieser Vorgehensweise: die prädizierten Schätzfehlerkovarianzen vermindern sich nach einem K-Band-Update nur unwesentlich im Vergleich zum Haupt-Update bei 60 Sekunden. Dies ist dadurch bedingt, dass bei jenem nur eine relative (skalare) Messgröße, bei diesem jedoch zudem eine direkte Beobachtung des Systemzustands eintrifft. Würde man also die nach Gl. (4.3) berechnete Matrix `Q` bei jedem 5-Sekunden-Prädiktionsschritt auf alle prädizierten Schätzfehlerkovarianzmatrizen addieren, erhielte man am Endpunkt des 60-Sekunden-Intervalls unverhältnismäßig große Kovarianzen. Im Grunde könnte man auch die Matrix $\boldsymbol{Q}_k$ bei den 5-Sekunden-Prädiktionen auslassen und sie erst am Ende, d.h. vor dem nächsten Haupt-Update, auf alle prädizierten Kovarianzmatrizen des vergangenen 60-Sek.-Intervalls addieren – die Filtergleichungen sehen aber nun einmal die Addition der Matrix $\boldsymbol{Q}_k$ bei jedwedem Prädiktionsschritt vor. Deshalb wird $\boldsymbol{Q}_k$ hier durch zwölf (Anzahl der 5-Sek.-Prädiktionsschritte) geteilt, mit der Folge, dass die resultierenden Schätzfehlerkovarianzen am Anfang des 60-Sekunden-Zeitfensters zwar kleiner sind als im obigen Fall (`ekf_orbit` und `cov_sel = 'const'`), jedoch am Ende, vor dem nächsten Haupt-Update, in etwa ebenso groß sind.

Bei den anderen Methoden (`'lin'` oder `'ric'`) wird das Systemrauschen mithilfe der Spektraldichtematrix modelliert, die quasi die zeitliche Änderung des Systemrauschens beschreibt. `Q` wird in diesem Fall durch die Zeitdauer zwischen den Hauptkorrekturen (60 Sekunden) dividiert, da die Matrix in die Integration involviert ist (Variante `'ric'`) bzw. gemäß Gl. (3.51) mit dem aktuellen Zeitpunkt $i \cdot \delta t$ multipliziert wird (Variante `'lin'`):

$$\texttt{Q} = \frac{1}{\Delta t_{\text{nav}}} \begin{pmatrix} \texttt{qpos}^2 \cdot \mathbf{I}_{(3\times3)} & \mathbf{0}_{(3\times3)} \\ \mathbf{0}_{(3\times3)} & \texttt{qvel}^2 \cdot \mathbf{I}_{(3\times3)} \end{pmatrix} \tag{4.5}$$

Der Zeitpunkt $i\cdot\delta t$ entspricht dem i-ten Element des Zeitvektors `tspan`, der die Prädiktionszeitpunkte relativ zum Anfangszeitpunkt t_k beinhaltet, d.h. er unterteilt das Prädiktionsintervall $[t_k, t_{k+1}]$ in m gleiche Zeitabschnitte δt, beginnend bei 0 Sekunden. Seine Einträge ergeben sich demzufolge aus $\texttt{tspan}_{[i]} = i \cdot \delta t, \quad i = 0, \ldots, m$. In den Hauptfunktionen wird `tspan` auf diese Weise mit der zeitlichen Auflösung δt von einer Sekunde definiert und an `ekf_predict` und den jeweiligen Orbitintegrator weitergereicht. In der Funktion `ekf_orbit` ergibt sich hier aus den Datensätzen $m = 60$, d.h. `tspan` $\widehat{=} [0, 1, \ldots 60]$, und in `ekf_orbit2` ist hier $m = 5$, sodass `tspan` $\widehat{=} [0, 1, \ldots 5]$.

Durch die Gleichungen (4.3) bis (4.5) für `Q` ist nun sichergestellt, dass in allen Kombinationen

der zwei Hauptfunktionen mit den drei Methoden `'ric'`, `'lin'` und `'const'` der Anteil des Systemrauschens am Ende eines 60-Sekunden-Prädiktionsintervalls in etwa gleich groß ist, sodass von daher vergleichbare Ergebnisse bei der Hauptkorrektur erzielt werden können. Je nach gewählter Methode wachsen aber eben die Schätzfehlerkovarianzen von einem Korrekturschritt zum nächsten unterschiedlich an.

Wie in Abb. 4.3 dargestellt, sind in allen numerischen Orbitintegratoren gewöhnliche DGL-Systeme zu lösen. An dieser Stelle würde es zu weit führen, die mathematische Theorie zur numerischen Lösung gewöhnlicher DGLen ausführlich darzulegen. Hierzu möge man entweder die entsprechende Fachliteratur, z.B. Stoer und Bulirsch (2005, Kap. 7), Reinhardt (2008), Hermann (2004), Henrici (1962) zu Rate ziehen oder, vor dem Hintergrund der Satellitenbahnberechnung, beispielsweise die Arbeiten von Scheinert (1996, S. 45 ff.) und Götzelmann (2003, Kap. 3).

Statt eigene Integratoren zu entwickeln, verwendet man hier einen der bereits in Matlab eingebauten Differentialgleichungslöser („ordinary differential equation (ode) solver"). Da es sich dabei um äußerst komplexe Lösungsalgorithmen handelt, kann hier in erster Linie nur deren Anwendung beschrieben werden. Weiterführende Informationen zu den Matlab-Integratoren findet man z.B. in Shampine und Reichelt (1997) sowie in Shampine et al. (2003).

Aus der besagten Theorie ist bekannt, dass der Verfahrensfehler des numerischen Integrationsverfahrens umso geringer ausfällt, je kleiner man die Integrationsschrittweite wählt. Dies gilt jedoch nur bis zu einer gewissen Schranke, unterhalb derer die Rundungsfehler überwiegen. Bei den Matlab-Integratoren wird die Schrittweite quasi automatisch ermittelt. Zufolge Schmidt (1999, S. 35) und der Matlab-Dokumentation zu `odeset` wird sie so klein gewählt, dass die Bedingung

$$|e_i| \leq \max\left\{\texttt{RelTol} \cdot |y_i|,\ \texttt{AbsTol}_{[i]}\right\} \tag{4.6}$$

eingehalten ist. Darin bezeichnet e_i den lokalen Diskretisationsfehler (auch Quadratur-/Verfahrensfehler) der i-ten Komponente des Lösungsvektors $\boldsymbol{y}$. Der DGL-Löser schätzt e_i für alle Komponenten i anhand einer für das jeweilige Integrationsverfahren gültigen Formel bei jedem Integrationsschritt ab. Ferner bezeichnen `RelTol` und `AbsTol` zulässige Toleranzwerte, die man mittels `odeset` verändern kann. Der skalare Wert `RelTol` stellt eine relative Fehlertoleranz dar, die praktisch die Anzahl der korrekten Nachkommastellen aller Lösungskomponenten kontrolliert. Zieht man in Erwägung, dass bei GRACE-Bahnen die Komponenten des Ortsvektors typischerweise maximale Werte in der Größenordnung von 7000 km erreichen, ist mit der unten gewählten relativen Toleranz von $\texttt{RelTol} = 10^{-13}$ wegen der Forderung (4.6) sichergestellt, dass der geschätzte Verfahrensfehler für die Positionskomponenten stets unter 0.7 µm liegt. Für die Geschwindigkeitskomponenten müsste analog der Fehler e kleiner sein als 0.8 nm/s, wenn man berücksichtigt, dass die Beträge der Geschwindigkeiten Maximalwerte um 8 km/s annehmen. Falls die absolute Toleranz `AbsTol` nicht falsch gewählt wird, dürften die Integrationsfehler somit sehr gering ausfallen. Der absolute Schwellenwert `AbsTol` gibt im Grunde an, wie groß der Fehler e maximal sein darf, wenn die Lösungskomponente einen Wert nahe Null annimmt. Denn in Gl. (4.6) ist in der Regel für Werte $y_i \neq 0$ das Produkt $\texttt{RelTol} \cdot |y_i|$ wesentlich größer als `AbsTol`. Man kann `AbsTol` entweder als skalare Größe (dann gilt der Wert für alle Komponenten) oder als Vektor definieren mit der gleichen Dimension wie `y0` (dann bezieht sich jeder Eintrag auf die zugehörige Komponente). Hier wurden für die Toleranzparameter in den Funktionen zur Orbitintegration die skalaren Werte

$$\texttt{RelTol} = 10^{-13}, \qquad \texttt{AbsTol} = 10^{-16}$$

gewählt. Andere Untersuchungen zur Satellitenbahnberechnung haben gezeigt, dass damit hinreichend genaue Ergebnisse bei vertretbarem Rechenaufwand erzielt werden können (s. z.B. König und Seitz, 2003, Kap. 6.1.2, S. 39).

Hinsichtlich der Wahl des Integrators empfiehlt die Matlab-Dokumentation für Anfangswertprobleme mit mittleren bis hohen Genauigkeitsanforderungen die Funktionen `ode45` und `ode113`. Die

erstgenannte Routine basiert auf expliziten Runge-Kutta-Verfahren vierter und fünfter Ordnung. Derartige Einschrittverfahren benötigen bekanntlich zur Berechnung der nächsten Lösung $y_i(t_{j+1})$ nur die Lösung des jeweils letzten Zeitpunkts $y_i(t_j)$. Demgegenüber handelt es sich bei `ode113` um ein Prädiktor-Korrektor-Verfahren nach Adams-Bashforth/Adams-Moulton und gehört damit zur Klasse der Mehrschrittverfahren, die sich dadurch auszeichnen, dass sie auf zwei (oder mehr) Näherungswerte vorhergehender Zeitpunkte $(y_i(t_j), y_i(t_{j-1}), \ldots)$ zurückgreifen, um den Näherungswert am darauffolgenden Punkt $y_i(t_{j+1})$ zu ermitteln. Nachteilig beim Einschrittverfahren ist unter anderem, dass die rechte Seite der DGL sehr oft ausgewertet werden muss, während dies beim Mehrschrittverfahren nur einmalig bei jedem Integrationsschritt erforderlich ist. So haben König und Seitz (2003) durch Tests demonstriert, dass die numerische Integration der Satellitenbahn mittels `ode113` unter Umständen wesentlich effizienter funktioniert; in einem konkreten Beispiel hat demnach `ode45` bei vergleichbarem Genauigkeitsniveau ungefähr die zehnfache Anzahl an Integrationsschritten benötigt. Auch Schmidt (1999, S. 55) legt nahe, zur Integration leicht elliptischer Bahnen mit $e < 0.3$ die Funktion `ode113` zu nutzen[1]. Seinen Tests zufolge liefert der alternative Integrator `ode45` nur bei stärker elliptischen Bahnen etwas bessere Ergebnisse – aber eben bei deutlich längeren Rechenzeiten. Aus diesen Gründen wurde auch hier die Funktion `ode113` für die Orbitintegration bevorzugt.

Im Folgenden wird die Implementierung des Orbitintegrators `orbint_field` näher betrachtet, stellvertretend für alle anderen programmierten numerischen Orbitintegratoren. Die Ein- und Ausgabevariablen der Orbitintegratoren und weiterer Funktionen sind dem Anhang A.2 bzw. den Hilfstexten der Quellcodes zu entnehmen.

Nach einigen Vorabprüfungen zur korrekten Eingabe der Eingangsvariablen wird in der Funktionorbint_field im Fall der Variante `'ric'` zunächst der Vektor mit den Anfangsbedingungen `invec` aus den Eingangsvariablen aufgestellt – dies sind gerade der zuletzt geschätzte Systemzustand und dessen Kovarianzmatrix (bzw. zu Beginn die Werte aus der Filterinitialisierung). Nach Festsetzung der Toleranzparameter mittels `odeset` erfolgt der Aufruf des DGL-Lösers nach der vorgegebenen Syntax:

```
139  % input vector for ode function at t0
140  invec = [X0(1:6);P0]; % (42 x 1)
141
142  options = odeset('RelTol',1e-13,'AbsTol',1e-16);
143  [~,out] = ode113(@fun1,tspan,invec,options); % out: (n x 42)
144  X = out(:,1:6);
145  mat = out(:,7:end);
```

Der zurückgegebene Vektor `out` enthält die an den Zeitpunkten `tspan` prädizierten Werte und wird entsprechend aufgeteilt in die zwei Ausgangsvariablen der Funktion `orbint_field`: zum einen die Zeitreihe der Systemzustände (`X`), zum anderen hier im Fall `'ric'` die Zeitreihe der zugehörigen Kovarianzmatrizen (`mat`). Die letzte Zeile dieser Ausgangsvariablen geht dann in den darauffolgenden Korrekturschritt als $\bar{\boldsymbol{s}}_{k+1}$ bzw. $\bar{\boldsymbol{P}}_{k+1}$ ein.

Damit `ode113` überhaupt diese Zeitreihen zurückliefern kann, muss die rechte Seite des DGL-Systems in Abhängigkeit vom Integrationszeitpunkt und den Funktionswerten in einer separaten Funktion (`odefun`) formuliert werden, welche dann vom Integrator ausgewertet werden kann. Dies geschieht hier durch die eingebettete Funktion `fun1`. Zuerst wird darin der Zustandsvektor `X0` und die Kovarianzmatrix `P0` aus dem Eingangsvektor `invec` aufgestellt, danach muss die für den Integrationszeitpunkt gültige Drehmatrix durch sphärische Interpolation aus der Zeitreihe `Ri2e` bestimmt werden (Funktion `slerp_mat`, s. Abschn. B.1 im Anhang):

```
170  function outvec = fun1(Delta_t,invec)
171  % system of ordinary differential equations as function handle for
172  % ode-solver, comprising the equations of motion and the Riccati ODE
173  % invec  = [rx, ry, rz, vx, vy, vz, P11, ... ,P66] (42x1)
174  % outvec = [vx, vy, vz, ax, ay, az, Pp11,...,Pp66] (42x1)
175
```

1 Bei GRACE-Orbits liegt die Exzentrizität, wie in Abschn. 2.1.1 erwähnt, typischerweise im Bereich von 0.001.

```
176 % extract state vector & covariance matrix from invec
177 X0 = invec(1:6);
178 P0 = reshape(invec(7:end),6,6)';
179
180 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
181 % equations of motion
182 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
183
184 % (1x9) rotation matrix Ri2e_i at time point of integration
185 if Delta_t==0
186     Ri2e_i = Ri2e(1,:);
187 elseif Delta_t==tspan(end)
188     Ri2e_i = Ri2e(end,:);
189 else
190     Ri2e_a = reshape(Ri2e(floor(Delta_t/fs)+1,:),3,3)';
191     Ri2e_b = reshape(Ri2e(floor(Delta_t/fs)+2,:),3,3)';
192     Ri2e_i = mat2vec(slerp_mat(Ri2e_a,Ri2e_b,mod(Delta_t,fs)/fs,1e-008))';
193 end
```

Anschließend rotiert man die im raumfesten System gegebenen Anfangspositionen $\boldsymbol{X}_0$ mithilfe der sphärisch interpolierten Rotationsmatrix $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}$ in das erdfeste System (Funktion `rot_statevec`, s. Abschn. B.2 im Anhang), transformiert die kartesischen erdfesten Koordinaten in sphärische und übergibt diese zusammen mit dem bis zum gewählten Maximalgrad begrenzten a priori Gravitationsfeld `field` an die Funktion `gradpshssp`, welche daraus gemäß den Formeln in Abschnitt 3.2.5.1 den Gravitationsgradienten in kartesischen Koordinaten bezüglich des erdfesten Systems ermittelt:

```
195 % coordinates in Earth-fixed spherical frame and gradient
196 [pos_e] = rot_statevec('i2e',Ri2e_i,X0(1:3)');
197 [lam,phi,r] = cart2sph(pos_e(1),pos_e(2),pos_e(3));
198 gradV       = gradpshssp(field,lam,phi,r,[],[],'xyz',0);
```

Im Vorbereitungsabschnitt dieser Funktion `gradpshssp` wird zunächst die eingelesene Matrix `field`, welche die Potentialkoeffizienten enthält, gegebenenfalls mittels der Funktion `cs2sc` vom CS-Format in das SC-Dreiecksformat überführt[1] und auf den gewählten Maximalgrad `lmax` herabgesetzt. Anschließend werden sämtliche Terme der Reihenentwicklungen für die ersten partiellen Ableitungen des Gravitationspotentials (Gl. (3.67) bis (3.69)) in demselben Format zwischengespeichert. Die Legendre-Funktionen (`p`) und deren erste Ableitungen (`dp`) werden dabei in der Funktion `plmsp` für die gegebenen sphärischen Koordinaten rekursiv berechnet. Die Ausgabevariablen `p` und `dp` sind ebenfalls Matrizen im SC-Format:

```
108 th   = pi/2 - phi;                    % get co-latitude
109 l    = (0:lmax)';
110 m    = l';
111 updw = (ae./r).^l;
112 l    = l(:,ones(1,2*lmax+1));
113 updw = updw(:,ones(1,2*lmax+1));
114 ml   = m.*lam;
115 sml  = sin(ml);
116 cml  = cos(ml);
117 scml = ones(lmax+1,1)*[fliplr(sml(2:end))  cml];
118 csml = ones(lmax+1,1)*[fliplr(m(:,2:end).*cml(:,2:end)) -m.*sml];
119
120 % fully normalized Legendre functions and first derivatives at point th
121 [p,dp] = plmsp(lmax,th*rho);
```

Das SC-Datenformat lässt sich auch als Spaltenvektor mit $(l_{\max}+1)\cdot(2\cdot l_{\max}+1)$ Einträgen schreiben, sodass man im Prinzip für jede der drei Reihenentwicklungen

$$\frac{\partial V}{\partial r}, \quad \frac{\partial V}{\partial \varphi}, \quad \frac{\partial V}{\partial \lambda}$$

einen Spaltenvektor bilden kann, der alle einzelnen, in der Entwicklung relevanten Summanden enthält. Mittels des Befehls `sum` können dann einfach alle Einträge dieses Spaltenvektors aufsummiert werden, um die jeweilige sphärische Komponente des Gravitationsgradienten im lokalen

1 Zur Erläuterung der Speicherformate der Potentialkoeffizienten siehe `csformats.pdf` auf der beigefügten CD.

Horizontsystem zu erhalten:

```
123  field = field(:);
124  scml  = scml(:);
125  p     = p(:);
126  updw  = updw(:);
127
128  gradV(1,1) = -GM./r.^2 .* sum( updw .* (l(:)+1) .* field .* scml    .*  p   );
129  gradV(2,1) =   GM./r    .* sum( updw .*              field .* scml    .* dp(:));
130  gradV(3,1) =   GM./r    .* sum( updw .*              field .* csml(:) .*  p   );
```

Zuletzt wird in **gradpshssp** noch der Gravitationsgradient durch die Vorschrift (3.70) in das erdfeste, kartestische Bezugssystem transformiert.

Zurück in der Unterfunktion **fun1** der Funktion **orbint_field** überführt man die Beschleunigungen **gradV** mit der Inversen der interpolierten Rotationsmatrix $\boldsymbol{R}_{\mathrm{I}}^{\mathrm{E}} = (\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}})^{-1} = (\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}})^{\top}$ in das raumfeste Bezugssystem:

```
200  % (3x3) rotation matrix Re2i_i
201  Re2i_i = reshape(Ri2e_i,3,3); % approx. equal to inverse of Ri2e
202
203  % accelerations in quasi-inertial frame
204  acc_i = Re2i_i * gradV(:);
```

Damit stehen nun die ersten sechs Komponenten des DGL-Systems (bzw. der Ausgangsvariable **outvec**) fest. Die ersten drei DGLen für den Zustandsvektor sind einfach die Geschwindigkeitskomponenten des Eingangsvektors **invec**. Der noch fehlende, aus 36 Einträgen bestehende zweite Teil des DGL-Systems ist die rechte Seite der Riccati-DGL. Dazu ist zuerst der Marussi-Tensor bezüglich des raumfesten Systems zu ermitteln. Der Tensor im erdfesten System ergibt sich entweder (falls **'PetVer'** gewählt wurde) basierend auf dem Ansatz nach Petrovskaya und Vershkov mithilfe der Funktion **tenspshs** oder mithilfe der Funktion **tenssp**:

```
210  % Marussi tensor matrix w.r.t. quasi-inertial system
211  if strcmp(tens_sel,'PetVer') % solution acc. to Petrovskaya/Vershkov
212      tensVe = tenspshs(field2,lam,phi,r,[],type,'xyz',[],0).*1e-9; % [1/s^2]
213      tensVe = reshape(tensVe,3,3)';
214      tensVi = tens_kep(X0) + Re2i_i*tensVe*(Re2i_i)';
215  else % default solution
216      tensVe = tenssp(field2,lam,phi,r,[],type,'xyz',0); % units: [1/s^2]
217      tensVi = Re2i_i*tensVe*(Re2i_i)';
218  end
```

Da **tenspshs** den Tensor nur für Grade ≥ 2 liefert, muss man den Tensor für Grad 0 (Kepler-Fall) hinzuaddieren, der hier in der Unterfunktion **tens_kep** bestimmt wird, wobei lediglich Gleichung (3.102) umgesetzt wurde. Die Funktion **tenssp** (und **tenspshs**) ist durch Adaption der Funktion **gradpshssp** entstanden. Hier gehen viele Berechnungen in analoger Weise vonstatten wie sie bereits oben für **gradpshssp** beschrieben wurden. Der wesentliche Unterschied besteht darin, dass nun **plmsp** auch die zweiten Ableitungen der Legendre-Funktionen liefern muss und neben den Reihenentwicklungen der ersten Ableitungen des Gravitationspotentials auch die Reihenentwicklungen der zweiten Ableitungen (Gl. (3.73) bis (3.78)) anzuwenden sind, s. Quelltext der Funktion **tenssp**. Zur Bestimmung des Marussi-Tensors sind also ungleich mehr Ableitungen und Summationen durchzuführen, sodass sich die Rechenzeiten bei der Integration drastisch verlängern würden, falls man hierbei denselben Entwicklungsgrad **lmax** ansetzt wie bei der Berechnung des Gravitationsgradienten mithilfe die Funktion **gradpshssp**. Daher ist es sinnvoll, den Marussi-Tensor nur bis zu einem geringeren Grad **lmax2** zu entwickeln, zumal für die Prädiktion der Schätzfehlerkovarianzmatrix (wofür man den Tensor ausschließlich verwendet) geringere Genauigkeitsansprüche gelten können als für die Prädiktion des Systemzustandes. Denn die Schätzfehlerkovarianzen sind in der Regel klein, sodass sich eventuell auftretende Prädiktionsfehler nur gering auswirken. Der Zustandsvektor hat hingegen vergleichsweise große Einträge, weshalb hier eher Wert auf eine möglichst präzise Voraussage zu legen ist. Im Einstellungsfenster des Hauptprogramms **ekf_orbit_main** ist für den Gravitationsgradient der Maximalgrad 60 und für den Marussi-Tensor der Maximalgrad 10 voreingestellt (s. Abb. 4.2, Optionen ③ und ④). Bei der Verarbeitung realer

Daten hat sich gezeigt, dass mit höheren Graden `lmax` die Differenzen zwischen prädizierten und beobachteten Bahnvektoren kaum geringer ausfallen. Der Wert für `lmax2` hat sich in numerischen Tests (Abschn. 4.6) als hinreichend zur Prädiktion der Schätzfehlerkovarianzen erwiesen. In der Unterfunktion `fun1` (s. obiger Quellcode-Auszug) wird an die Funktionen `tenssp` bzw. `tenspshs` das bereits vorab bis zum Grad `lmax2` reduzierte Gravitationsfeld `field2` übergeben.

Am Ende der Funktion `tenssp` stellt man aus den partiellen Ableitungen von V die Matrix $\boldsymbol{M}_{\mathrm{L}}$ (Marussi-Tensor bezüglich des lokalen Horizontsystems) gemäß Gleichung (3.72) auf und transformiert $\boldsymbol{M}_{\mathrm{L}}$ nach Vorschrift (3.79) in das erdfeste Referenzsystem. Somit erhalten wir in `fun1` den raumfesten Tensor `tensVi` durch die oben angegebene, einfache orthogonale Transformation `Re2i_i*tensVe*(Re2i_i)'` mit der interpolierten Rotationsmatrix `Ri2e_i`. Gemäß Gl. (3.56) lässt sich mithilfe des raumfesten Marussi-Tensors auch sofort die Jacobi-Matrix $\boldsymbol{F}$ angeben:

```
220 % Jacobi matrix
221 F = [zeros(3,3) , eye(3,3);
222      tensVi   , zeros(3,3)];
```

Im Anschluss ergibt sich die zeitliche Ableitung der Kovarianzmatrix $\dot{\boldsymbol{P}}$ zufolge der Riccati-DGL (3.36) unter Verwendung der soeben bestimmten Jacobi-Matrix, der zu den Eingangsvariablen gehörenden Matrix `Q` sowie der aus `invec` extrahierten Kovarianzmatrix `P0`:

```
224 % Riccati ODE system
225 Pp = F*P0 + P0*F' + Q;
```

Damit kann schließlich der Ausgabevektor `outvec` aufgestellt werden, welcher die gesamte rechte Seite des DGL-Systems für die Variante `'ric'` repräsentiert:

```
227 % export time derivatives to outvec
228 outvec = [invec(4:6);acc_i;mat2vec(Pp)];
```

Somit ist die Unterfunktion `fun1` vollständig implementiert und der `ode113`-Löser liefert als Resultat der numerischen Integration die gesuchte Zeitreihe der Zustandsvektoren (`X`) und der Kovarianzmatrizen (`mat`).

Bei der Variante `'var'` ist hingegen die Zeitreihe der prädizierten Transitionsmatrizen gesucht (Ausgangsvariable `mat`), weshalb die gesamte Orbitintegration in leicht abgewandelter Form zu programmieren ist. Wie man aus Gl. (3.44) weiß, ist die Anfangsbedingung für die Transitionsmatrix eine (6×6)-Einheitsmatrix. Dementsprechend ist beim Aufruf des DGL-Lösers der Vektor `invec` abzuändern:

```
148 % error state transition matrix Phi at t0
149 Phi0  = mat2vec(eye(6,6));
150
151 % input vector for ode function at t0
152 invec = [X0(1:6);Phi0];
153
154 options = odeset('RelTol',1e-13,'AbsTol',1e-16);
155 [~,out] = ode113(@fun2,tspan,invec,options); % out: (n x 42)
156 X = out(:,1:6);
157 mat = out(:,7:end);
```

Außerdem wird hier die Unterfunktion `fun2` statt `fun1` verwendet, um die rechte Seite des DGL-Systems zu beschreiben. Der Großteil der Funktion `fun2`, angefangen bei den Bewegungsgleichungen bis einschließlich der Bestimmung der Jacobi-Matrix, ist aber identisch zur Funktion `fun1` (s. Quelltext der Funktion `orbint_field`)[1]. Lediglich die mit der Jacobi-Matrix definierte Matrix-DGL weicht von dem in `fun1` festgelegten System ab. An die Stelle der Riccati-DGL treten in der Funktion `fun2` nämlich die Variationsgleichungen, sodass die letzten 36 Einträge des Vektors `outvec` anders zustande kommen:

1 Die ersten sechs Einträge von `outvec` sind bei beiden Varianten identisch, folglich unterscheiden sich die prädizierten Zustandsvektoren beider Varianten nicht voneinander, sondern nur die Ausgangsvariablen `mat`.

```
288 % variational equations
289 Phip = F*Phi0;
290
291 % export time derivatives to outvec
292 outvec = [invec(4:6);acc_i;mat2vec(Phip)];
```

Zusammenfassend sind in Abbildung 4.4 die wesentlichen (gemeinsamen) Berechnungsschritte (für den Standardfall via `tenssp`, d.h. `tens_sel` $\neq$ `'PetVer'`) dargestellt, die sowohl in der Unterfunktion `fun1` als auch in `fun2` implementiert sind. Da diese Berechnungen bei jedem Aufruf durch den DGL-Löser `ode113`, also bei jedem Integrationsschritt, erforderlich sind, ist der Rechenaufwand nicht gerade als gering einzustufen. Bei den anderen Orbitintegratoren, die auf einfacheren Kräftemodellen beruhen, verbessert sich die Situation in dieser Hinsicht in gewissem Maße, was auch anhand der in den nächsten Kapiteln ermittelten Rechenzeiten deutlich wird.

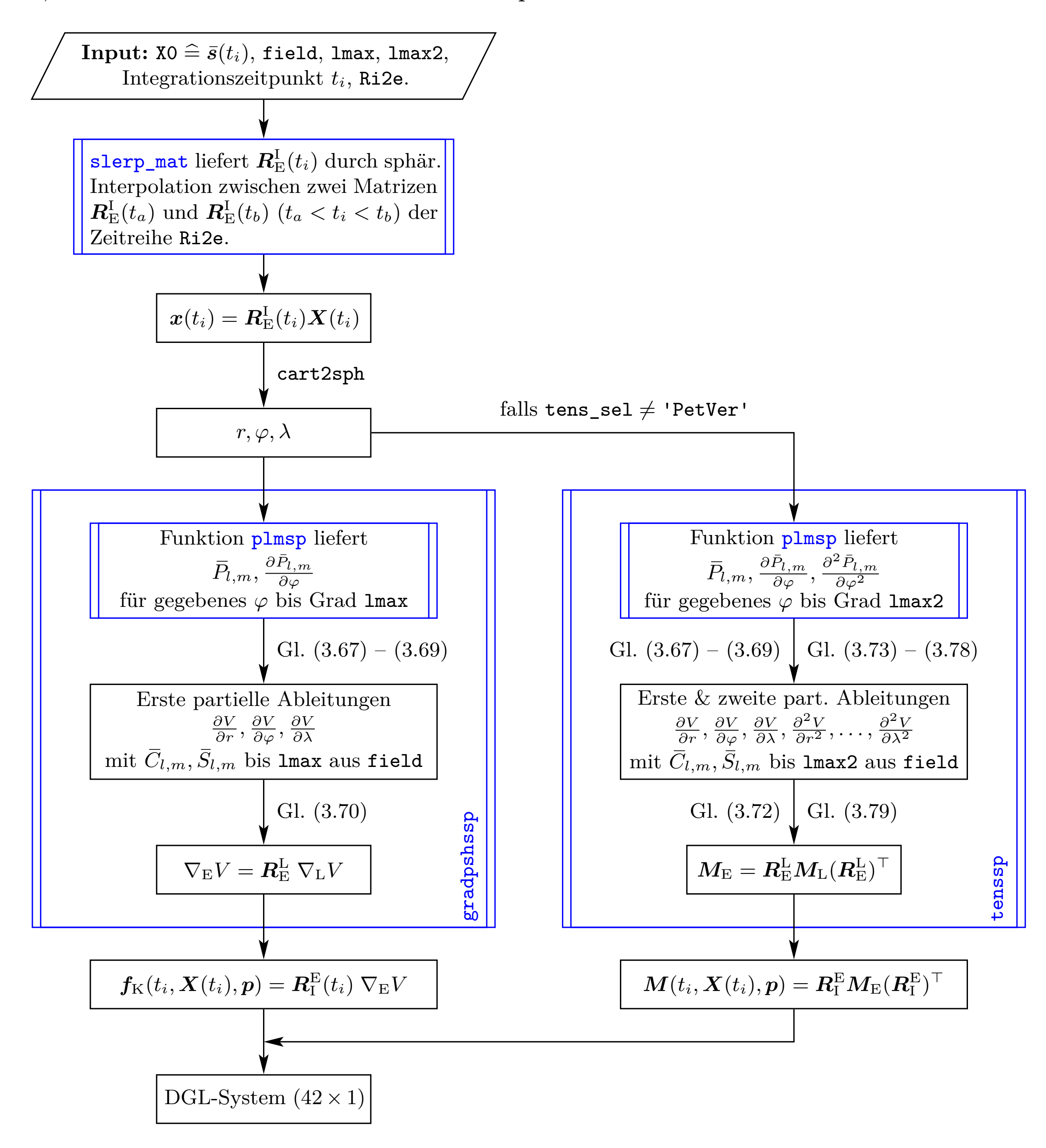


Abb. 4.4: Ablaufdiagramm zur Funktion `orbint_field`

Bis auf den Orbitintegrator `orbint_kep2`, der die Satellitenbahn größtenteils mithilfe geschlossen analytischer Formeln berechnet, wurden alle übrigen Orbitintegratoren nach dem oben vorgestellten Prinzip der numerischen Integration mit jeweils zwei Varianten `'ric'` und `'var'` und den

entsprechenden Unterfunktionen `fun1` und `fun2` umgesetzt. Außerdem ist jeweils noch eine weitere Unterfunktion `fun3` zu finden, die folgenden Zweck hat: falls der Nutzer einen Orbitintegrator separat, d.h. nicht zur Prädiktion innerhalb des Kalman-Filters nutzen möchte und nur am integrierten Orbit (Zeitreihe `X`) interessiert ist, daher also keine Ausgabe der Matrix `mat` verlangt (nur ein Ausgabeargument), wird der Orbit mithilfe dieser Unterfunktion `fun3` integriert, die nur die entsprechende Bewegungsgleichung nutzt. In diesem Fall kann somit Rechenzeit eingespart werden.

Die numerischen Integratoren unterscheiden sich im Grunde nur hinsichtlich der Kräftefunktion und des Marussi-Tensors, wie bereits mit den entsprechenden Formeln in Abschnitt 3.2.5 dargelegt wurde. Es wird daher darauf verzichtet, die Quellcodes dieser Funktionen hier ebenfalls explizit zu beschreiben.

Unter den Orbitintegratoren wurde der Zweck der Funktion `orbint_field2` bislang noch nicht erläutert: diese entspricht der Funktion `orbint_field`, nur dass der Marussi-Tensor nicht über die Funktion `tenssp` oder `tenspshs`, sondern nach den Formeln des J_2-gestörten Falles berechnet wird (wie in `orbint_j2`). Der Marussi-Tensor geht bekanntlich nur in die DGLen ein, die der Fehlerfortpflanzung dienen. Es werden also ausschließlich die *Kovarianzen* des Systemzustands nach dem J_2-Kräftemodell fortgeschrieben, wodurch sich der Rechenaufwand allerdings erheblich reduziert. Man erhofft sich dennoch, zu einem vergleichbar genauen Orbit zu gelangen. Denn wie bereits erwähnt, müsste in den meisten Fällen ein einfaches Kräftemodell genügen, um die Schätzfehlerkovarianzen zu prädizieren (aufgrund der betragsmäßig kleinen Werte). Probleme dürften erst dann auftreten, wenn die Kovarianzen durch die Vereinfachung derart ungenau prädiziert werden, dass in der Folge das Messungs-Update nicht mehr zuverlässig vonstatten geht. Ob sich diese Sorge hier bewahrheitet, zeigt sich unter anderem bei den noch folgenden numerischen Tests.

An dieser Stelle sei noch kurz die Implementierung des Orbitintegrators `orbint_kep2` erklärt. Darin berechnet man nach der Überprüfung der Eingangsvariablen zuerst die zum gegebenen Ausgangszustand zugehörigen Keplerelemente, indem man den ersten Schritt der in Tafel 2.3 skizzierten Ephemeridenrechnung durchführt:

$$\hat{\boldsymbol{s}}_0 = \left(\boldsymbol{X}_0, \dot{\boldsymbol{X}}_0\right)^\top \xrightarrow{\texttt{cart2kepler}} \{a, e, i, \omega, \Omega, M_0\}$$

Anschließend prädiziert man die mittlere Anomalie für alle mit `tspan` gegebenen Zeitpunkte gemäß

$$M(t_i) = M_0 + n \cdot i \cdot \delta t, \quad \text{wobei} \quad n = \sqrt{GM/a^3}.$$

In dem Fall, dass nur die Bahnvektoren auszugeben sind (keine Ausgangsvariable `mat`) sowie im Fall `'ric'` gewinnt man die zukünftigen Bahnvektoren mittels der Ephemeridenrechnung, d.h. die zuvor für jeden Zeitpunkt berechneten Keplerelemente überführt man (in einer Schleife) wieder in die zugehörigen Positions- und Geschwindigkeitsvektoren:

$$\{a, e, i, \omega, \Omega, M(t_i)\} \xrightarrow{\texttt{kepler2cart}} \left(\boldsymbol{X}(t_i), \dot{\boldsymbol{X}}(t_i)\right)^\top = \bar{\boldsymbol{s}}(t_i)$$

Im Fall `'ric'` erfolgt danach die Prädiktion der Schätzfehlerkovarianzen durch numerische Integration der Riccati-DGL, wobei die Positionen, die in den Marussi-Tensor nach Gl. (3.102) Eingang finden, allerdings nach der Ephemeridenrechnung gewonnen werden. In diesem speziellen Fall ist die Lösung daher eine gemischt analytisch-numerische mit 36 DGLen (siehe Unterfunktion `fun1`).

Bei der Methode `'var'` entfällt jegliche numerische Integration, da hier die analytische Lagrange/Gibbs-Lösung gemäß Abschnitt 3.2.5.4 (Seite 55) umgesetzt wurde. In der Funktion wird dazu der gegebene Zeitvektor `tspan` in einer Schleife abgearbeitet und die Formeln werden in jedem Durchlauf nur für einen Zeitpunkt ausgewertet, indem man aus der zuvor propagierten mittleren

Anomalie die exzentrische Anomalie durch iterative Auflösung der Keplergleichung

$$M(t) = E(t) - e \cdot \sin(E(t)) \tag{4.7}$$

in der Funktion **keplereq** ermittelt und diese zusammen mit E_0, den Keplerelementen a und e, der Zeitdifferenz $i \cdot \delta t$ sowie dem Anfangszustand $\hat{\boldsymbol{s}}_0$ an die Funktion **fgkepler** weitergibt, die daraus gemäß den Formeln in Abschn. 3.2.5.4 den prädizierten Zustandsvektor und die Transitionsmatrix für den momentanen Zeitpunkt berechnet:

```
148 for i=1:length(tspan)
149         E = sym(keplereq(double(vpa(e0,19)),double(vpa(M(i),19))));
150         [X(i,:),mat(i,:)] = fgkepler(X0,tspan(i),a0,e0,E0,E,type,'num');
151 end
```

Für viele Berechnungen in der Funktion **orbint_kep2** und den untergeordneten Funktionen **cart2kepler**, **kepler2cart** und **fgkepler** nutzt man die Symbolic Math Toolbox™, um Rundungsfehler klein zu halten. Ohne diese Toolbox hatten zuvor die Rundungsfehler den im nächsten Abschnitt folgenden Vergleich mit der numerischen Lösung via **orbint_kep** zu stark gestört. **orbint_kep2** ist nur der Vollständigkeit halber im Optionen-Fenster des Hauptprogramms **ekf_orbit_main** auswählbar. Da die Symbolic Math Toolbox™ hier extrem lange Rechenzeiten verursacht, ist sie zur Integration von längeren Stunden- oder Tagesorbits nicht zu empfehlen.

4.5 Test der Orbitintegratoren mithilfe simulierter Daten

Um sicherzugehen, dass die implementierten Programme bzw. Funktionen keine groben Fehler aufweisen, sondern brauchbare und im Rahmen der Rechengenauigkeit tolerierbare Ergebnisse liefern, ist es unerlässlich, diese durch geeignete Testrechnungen zu kontrollieren. Nebenbei erhält man durch die Messung und den Vergleich der Rechenzeiten Auskunft über die Effizienz bestimmter Funktionen und Berechnungen bzw. über mögliche Effizienzdefizite. Sämtliche Berechnungen wurden auf einem PC mit den in Tabelle 4.2 angegebenen Systemspezifikationen durchgeführt. Die genannten Rechenzeiten sind also stets in Relation zu einem derartigen Computersystem zu sehen.

Tab. 4.2: Kenndaten der genutzten Hard- und Software

MATLAB®-Version	7.10.0.499 (R2010a), 32 bit
Betriebssystem	Windows® 7 Professional SP1 32 bit
Prozessor	Intel® Core™ 2 Duo P9600 @ 2.67 GHz
Arbeitsspeicher	4 GB DDR3, 532.0 MHz (1:2)

In diesem Abschnitt folgen nun zuerst einige Tests der Funktionen, die für die Orbitintegration zuständig sind. Diese Untersuchungen sind insbesondere deshalb wichtig, weil eine fehlerhafte Prädiktion, wie schon angemerkt, die Divergenz des Kalman-Filters zumindest begünstigen kann. Sämtliche Tests erfolgen im Programm `orbint_test_main`, welches entsprechend den nun folgenden Unterabschnitten gegliedert ist.

4.5.1 Numerische vs. analytischer Lösung des Keplerproblems

Da für das Keplerproblem numerische *und* analytische Lösungen möglich sind (und hier auch umgesetzt wurden), bietet sich dieses an, um die numerischen Integrationsverfahren zu verifizieren und darüber hinaus gegebenenfalls zu optimieren. Dabei vergleicht man jeweils die numerische mit der als Referenz angenommenen analytischen Lösung.

Als Anfangswerte für die Integrationen wurde folgender Bahnvektor der GRACE-A-Navigationslösung (Datensatz `52640.mat`, Zeitpunkt 0 s) gewählt:

$$\begin{pmatrix} \boldsymbol{X}_0 \\ \dot{\boldsymbol{X}}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_{\text{nav}}(t_0) \\ \dot{\boldsymbol{X}}_{\text{nav}}(t_0) \end{pmatrix} \doteq \begin{pmatrix} -2302987.136\,\text{m} \\ 2050322.019\,\text{m} \\ -6126813.614\,\text{m} \\ 4883.765084\,\text{m/s} \\ -4742.728582\,\text{m/s} \\ -3424.169900\,\text{m/s} \end{pmatrix} \tag{4.8}$$

Das im Testprogramm `orbint_test_main` angesetzte Integrationsintervall `tspan` von 0 bis 21600 s (6 Stunden) mit einem Zeitinkrement von 60 Sekunden (`tspan = 0:60:21600`) entspricht etwa 3.8 Umläufen eines GRACE-Satelliten, da dessen Umlaufzeit ca. 1 Stunde und 34 Minuten beträgt. Mit diesen Eingangsvariablen liefert `orbint_kep2` nach der Lagrange/Gibbs-Lösung den Referenzorbit und `orbint_kep` den numerisch integrierten Orbit. Bei der numerischen Lösung können nun die beiden Verfahren `ode113` und `ode45` sowie verschiedene Toleranzen getestet werden, die absolute Toleranz bleibt hier unverändert bei `AbsTol` $= 10^{-16}$, d.h. es wird nur `RelTol` variiert.

Mit dem Einschrittverfahren `ode45` und der Toleranz `RelTol` $= 10^{-13}$ ergeben sich die in Abb. 4.5 dargestellten Differenzen zwischen den Komponenten der integrierten Bahnvektoren. Sieht man von

den auftretenden Rundungsfehlern[1] ab und fasst man die analytische Lösung als fehlerfreie Referenz auf, spiegeln diese Abweichungen die globalen Verfahrensfehler der numerischen Integration wider. Die Abweichungen verringern sich im Ganzen um fast eine Größenordnung, wenn man die Toleranz für den lokalen Verfahrensfehler auf `RelTol` $= 2.3 \cdot 10^{-14}$ einschränkt (vgl. Abb. 4.6). In beiden Fällen steigen die Differenzen jedoch sukzessive mit der Integrationszeit an, da sich die lokalen Verfahrensfehler immer weiter aufsummieren.

Auch bei Verwendung des Mehrschrittverfahrens `ode113` zeigt sich das zeitabhängige, tendenziell ansteigende Fehlerverhalten (s. Abb. 4.7 und 4.8). Im Vergleich zum Einschrittverfahren liefert das Mehrschrittverfahren bereits mit dem Wert von 10^{-13} für den Toleranzparameter `RelTol` offenbar bessere Ergebnisse hinsichtlich der Integrationsfehler. Durch die strengere Toleranzvorgabe von $2.3{\cdot}10^{-14}$ lässt sich zwar insgesamt nochmals ein geringfügiger Genauigkeitsgewinn erreichen, jedoch kommt es bereits *früher* (nach ca. 1 1/2 Stunden) zu erkennbaren Abweichungen in Bezug auf die analytische Lösung (vorher, mit `RelTol` $= 10^{-13}$, ca. nach 2 Std.). Dies lässt sich auch Abb. 4.9 und 4.10 entnehmen, welche die Differenzen zwischen den analytisch und den numerisch bestimmten Transitionsmatrizen verzeichnen. Für diese Grafiken wurden jeweils nur die Einträge der rechten oberen (3×3)-Teilmatrix herangezogen, da sich hier die größten Abweichungen ergaben.

Die ermittelten Abweichungen der numerischen zur analytischen Lösung sind vollkommen zufriedenstellend, zumal die Integrationszeit beim Kalman-Filter aufgrund der häufigen Korrekturschritte, die hier in Abständen von 5 bzw. 60 Sekunden stattfinden, deutlich unterhalb von 6 Stunden liegt. Im ersten 60-Sek.-Zeitintervall liegen die berechneten Differenzen nur bei 1 nm bzw. 1 pm/s. Diese sind weniger auf die Fehler des Integrationsverfahrens sondern viel eher auf die Rundungsfehler bei den numerischen Berechnungen in MATLAB zurückzuführen.

Neben dem numerisch günstigeren Fehlerverhalten ist das Mehrschritt- dem Einschrittverfahren auch hinsichtlich der Rechenzeiten überlegen (s. Tab. 4.3). Damit bestätigt der Test, dass die Verwendung des `ode113`-Lösers mit dem relativen Toleranzparameter `RelTol` $= 10^{-13}$ (und dem Parameter `AbsTol` $= 10^{-16}$) in allen implementierten numerischen Orbitintegratoren eine gute Wahl darstellt.

Der Orbitintegrator `orbint_kep2` benötigt hier eine vergleichsweise lange Rechenzeit von etwa 18 Minuten. Dies ist vorwiegend der Symbolic Math Toolbox™ geschuldet. Sie wird für den größten Teil der Berechnungen dieser Funktion genutzt, da andernfalls die Rundungsfehler bei der Lösung nach den analytischen Formeln derart groß sind, dass sie die Verfahrensfehler der numerischen Integration gerade etwas übersteigen. In erster Linie dient die Funktion `orbint_kep2` aber ohnehin nur zur soeben erfolgten Kontrolle der numerischen Orbitintegration. Deswegen hat eine Steigerung der Berechnungsgeschwindigkeit dieser Funktion hier keine Prioriät.

Tab. 4.3: Rechenzeiten der Integratoren für einen 6-Stunden-Keplerorbit

Integrator	Typ	`RelTol`	Rechenzeit (Sek.)
`orbint_kep2`	analytisch	–	1081
`orbint_kep` & `ode45`	numerisch	$2.3 \cdot 10^{-14}$	1.9
`orbint_kep` & `ode45`	numerisch	10^{-13}	1.4
`orbint_kep` & `ode113`	numerisch	$2.3 \cdot 10^{-14}$	0.22
`orbint_kep` & `ode113`	numerisch	10^{-13}	0.17

1 Standardmäßig stellt MATLAB Zahlen (vom Typ „double precision“, 8 Byte) etwa auf 15 Dezimalstellen genau dar.

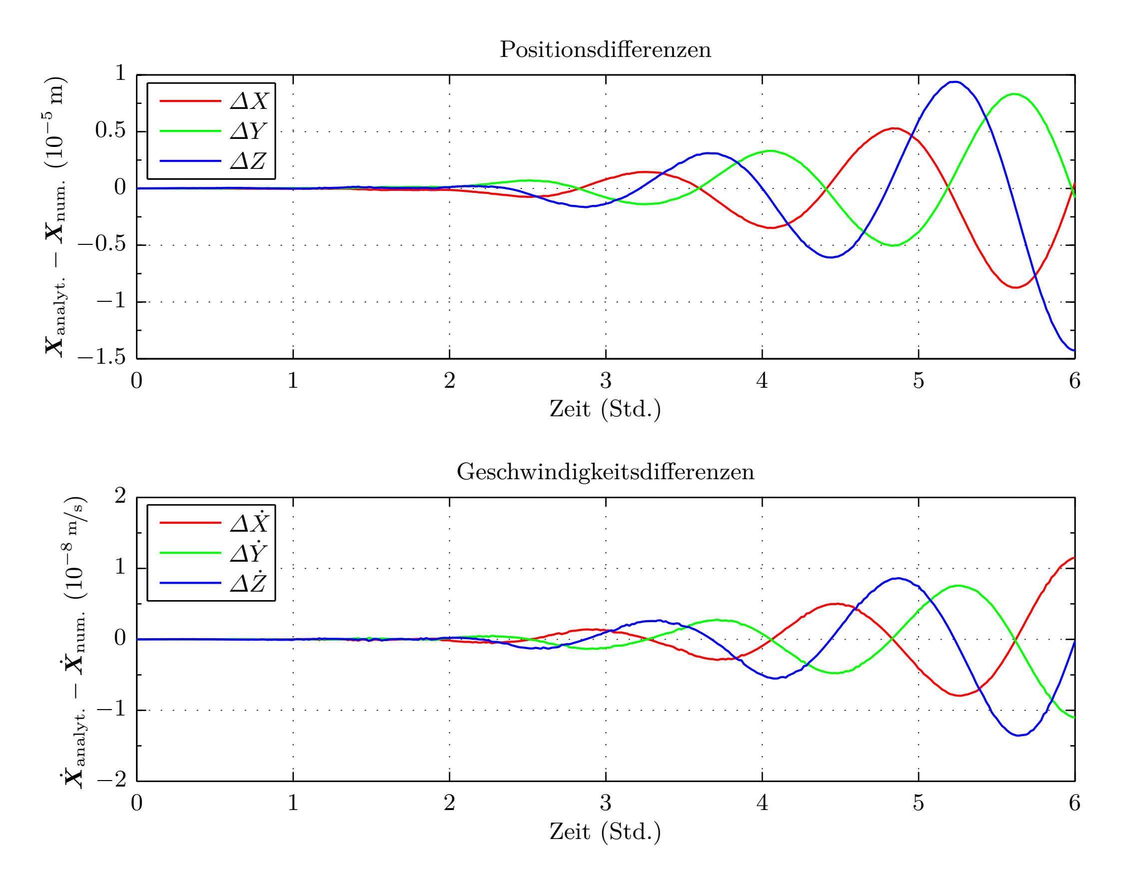


Abb. 4.5: Numerische (`ode45`, `RelTol` $= 10^{-13}$) vs. analytische Orbitbestimmung (Bahnvektordifferenzen)

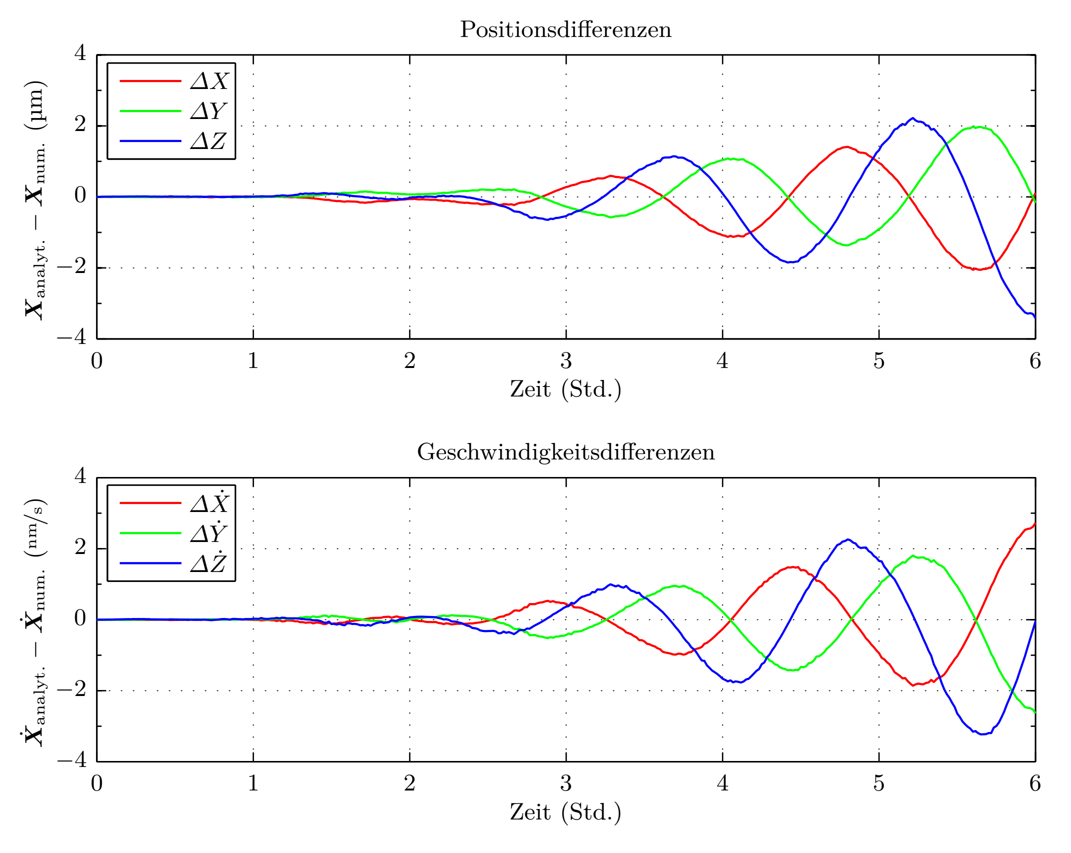


Abb. 4.6: Numerische (`ode45`, `RelTol` $= 2.3 \cdot 10^{-14}$) vs. analytische Orbitbestimmung (Bahnvektordifferenzen)

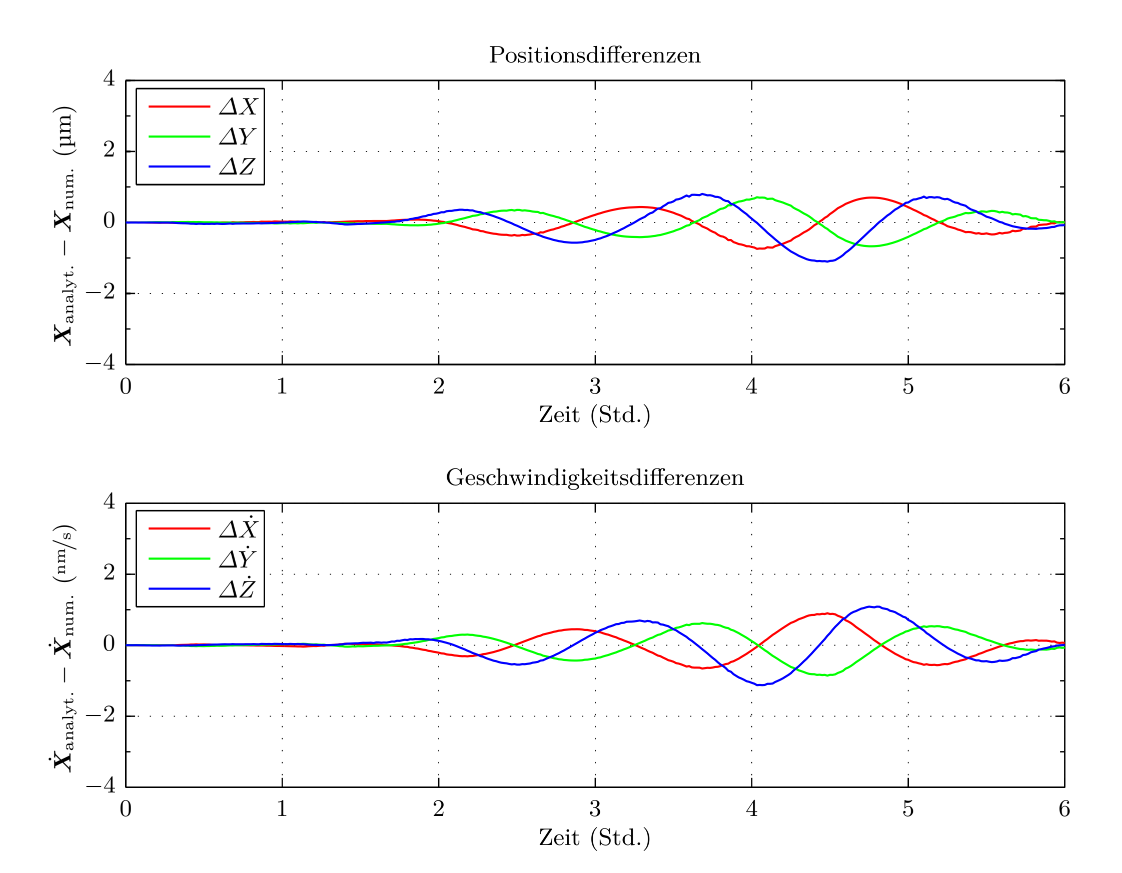


Abb. 4.7: Numerische (`ode113`, `RelTol` $= 10^{-13}$) vs. analytische Orbitbestimmung (Bahnvektordifferenzen)

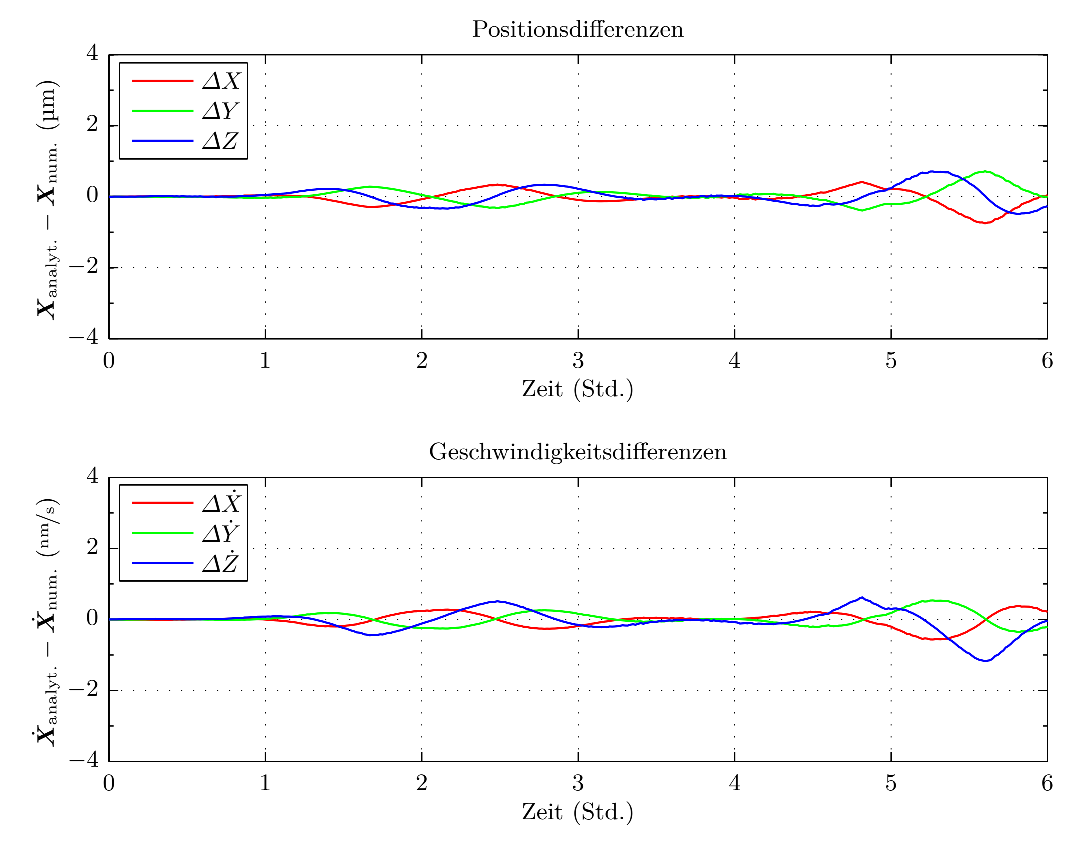


Abb. 4.8: Numerische (`ode113`, `RelTol` $= 2.3 \cdot 10^{-14}$) vs. analytische Orbitbestimmung (Bahnvektordiff.)

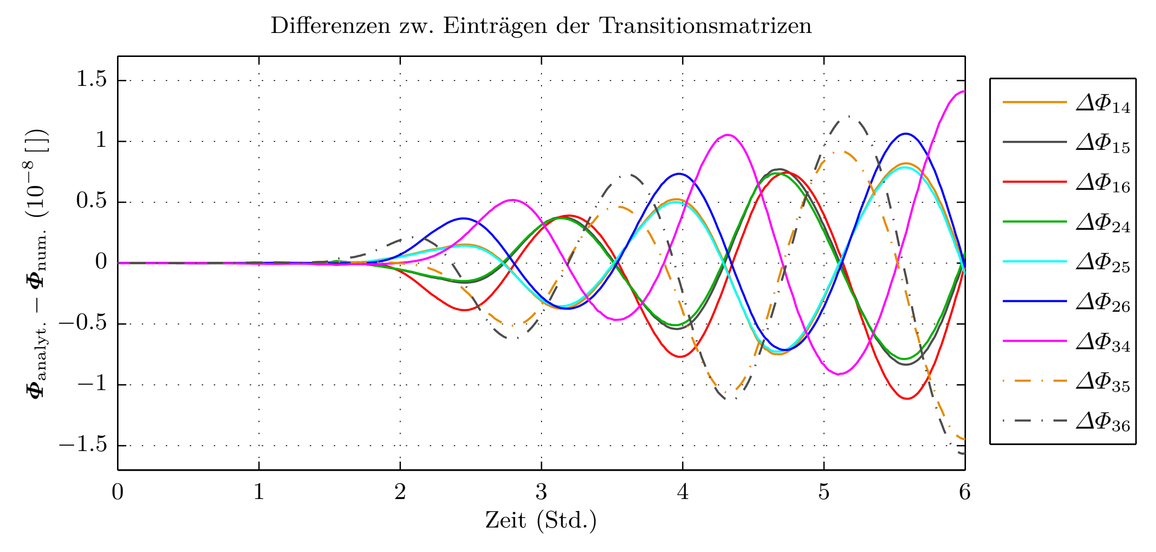


Abb. 4.9: Numerische (`ode113`, `RelTol` $= 10^{-13}$) vs. analytische Orbitbestimmung (Differenzen zwischen Transitionsmatrizen)

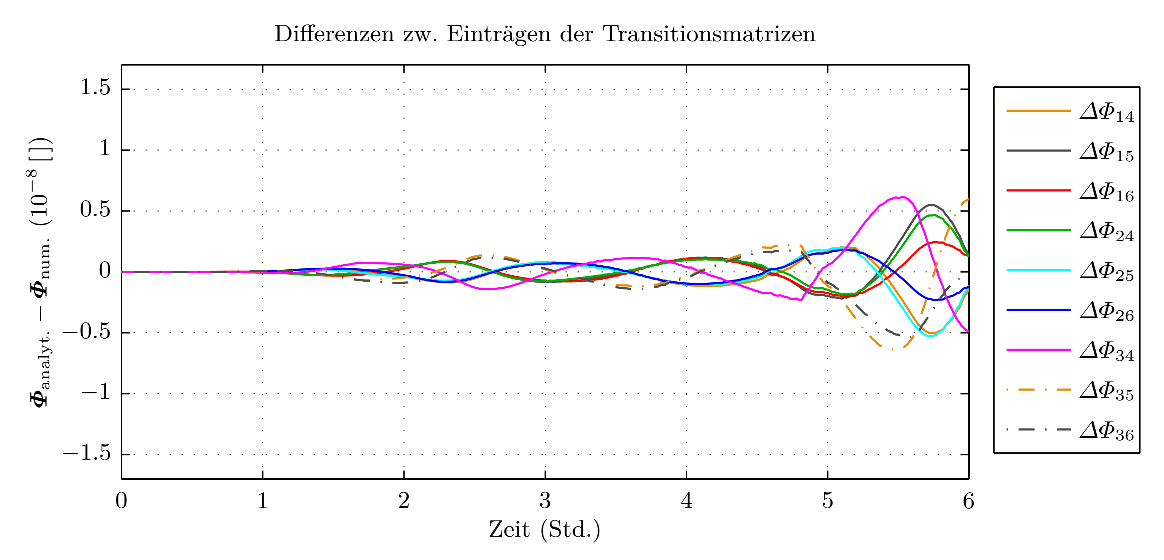


Abb. 4.10: Numerische (`ode113`, `RelTol` $= 2.3 \cdot 10^{-14}$) vs. analytische Orbitbestimmung (Differenzen zwischen Transitionsmatrizen)

4.5.2 Energie- und Drehimpulserhaltung im Kepler-Fall

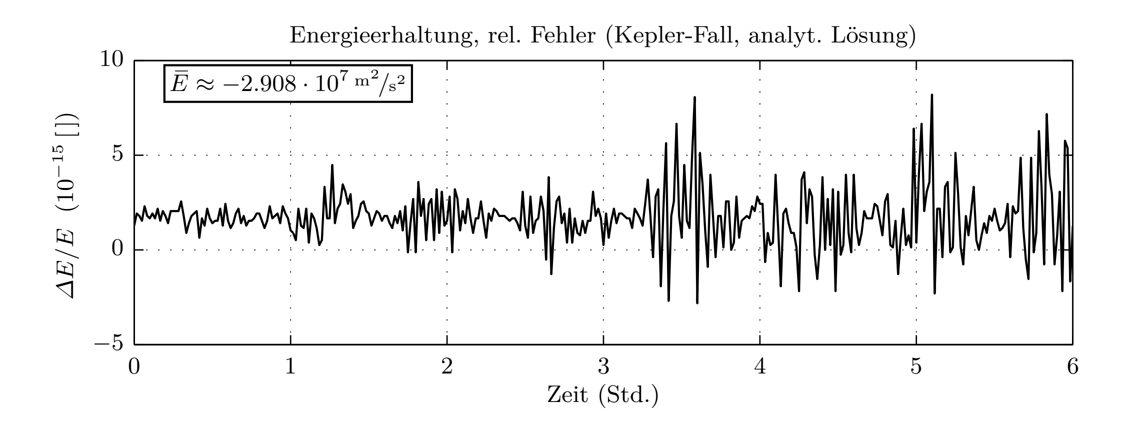
Im Keplerfall, wenn der Satellit also allein dem Einfluss eines kugelsymmetrischen Gravitationsfeldes ausgesetzt ist, müssen einige einfache physikalische Gesetzmäßigkeiten für den integrierten Orbit eingehalten werden, die sich ebenfalls durch Testrechnungen verifizieren lassen. Zum einen muss zufolge der Energieerhaltung die Gesamtenergie konstant sein, d.h. es muss für alle Bahnpunkte gelten (z.B. SEEBER, 1989, S. 72):

$$E = \frac{\|\dot{\boldsymbol{X}}\|^2}{2} - \frac{GM}{\|\boldsymbol{X}\|} = \text{const.}$$

SEEBER (1989, S. 72) zufolge muss darüberhinaus auch der Drehimpulsvektor $\boldsymbol{L} = \boldsymbol{X} \times \dot{\boldsymbol{X}}$ an jedem Zeitpunkt des integrierten Orbits die gleichen, konstanten Einträge haben. In Abbildung 4.11 und 4.12 sind die relativen Fehler

$$\frac{\Delta E(t)}{E(t)} = \frac{E(t) - \bar{E}}{E(t)} \quad \text{bzw.} \quad \frac{\Delta L_i(t)}{L_i(t)} = \frac{L_i(t) - \bar{L}_i}{L_i(t)} \; \forall \; i \in \{X, Y, Z\}$$

sowohl für den numerisch als auch für den analytisch berechneten Keplerorbit dargestellt. In beiden Fällen ist die Energie- und Drehimpulserhaltung mit hinreichender Genauigkeit erfüllt, denn die berechneten relativen Fehler in der Größenordnung von 10^{-15} bzw. 10^{-14} sind angesichts des besagten Rundungsfehlerniveaus als unkritisch einzustufen.



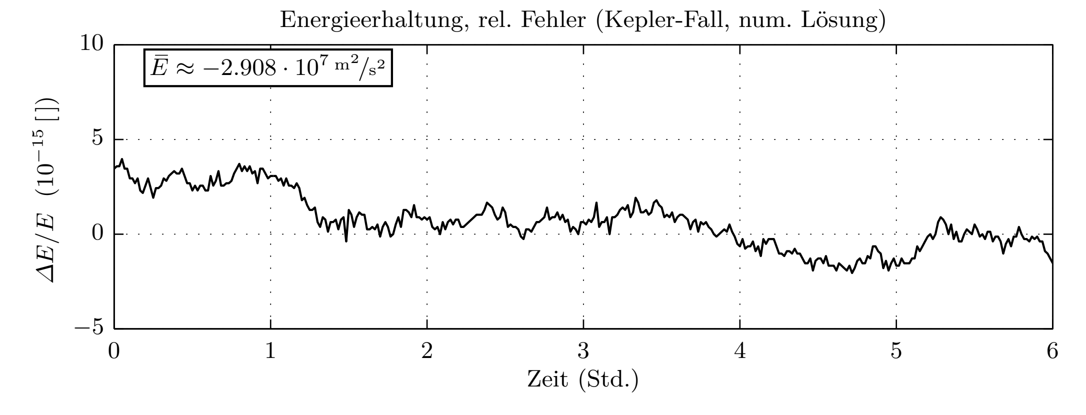


Abb. 4.11: Energieerhaltung der integrierten Keplerorbits

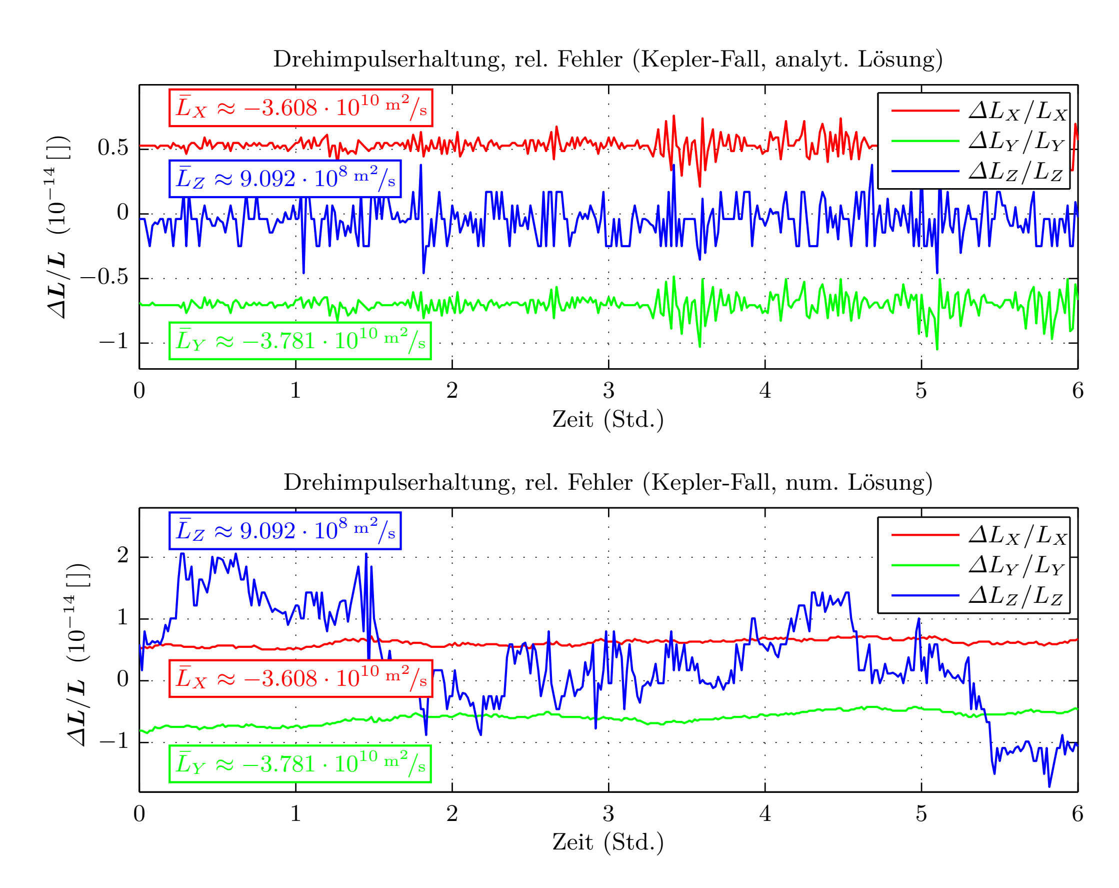


Abb. 4.12: Drehimpulserhaltung der integrierten Keplerorbits

4.5.3 Test der übrigen Integratoren mithilfe eines speziellen Kräftemodells

Für einen Orbitintegrator, dessen Kräftemodell auf einem zonalen Gravitationsfeld mit ebensolchen Potentialkoeffizienten (d.h. Ordnung $m = 0$) beruht, lässt sich ein Test auf die Konstanz der Z-Komponente des Drehimpulsvektors durchführen. Laut Schmidt (1999, S. 48) müssen dabei jedoch bestimmte Voraussetzungen erfüllt sein:

(i) Die felderzeugenden Massen sind fest mit dem erdfesten System verbunden und

(ii) rotieren *gleichförmig* mit der (konstanten) Winkelgeschwindigkeit $\boldsymbol{\omega} = (0, 0, \omega_\mathrm{E})^\top$ um die *gemeinsame* Z-Achse.

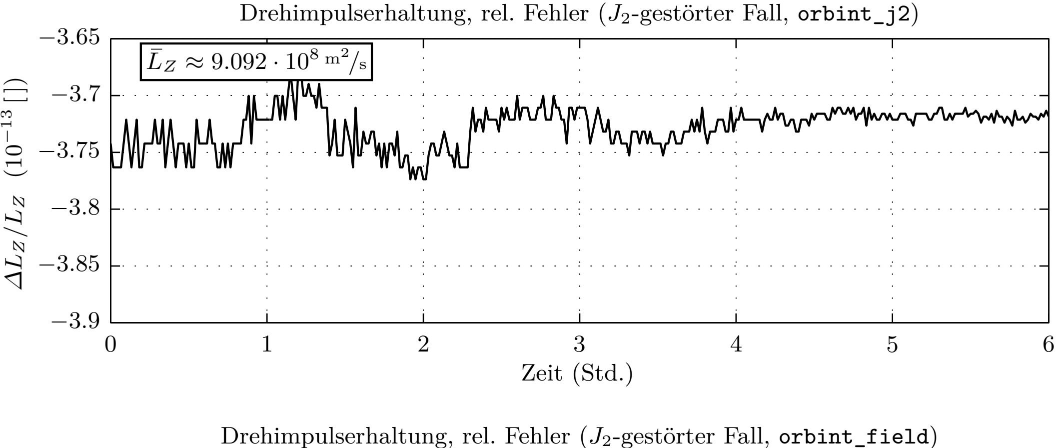
(iii) Die Potentialkoeffizienten des verwendeten Gravitationsfeldmodells sind zeitlich konstant.

Der Test ist demnach ohne weiteres mit dem Orbitintegrator `orbint_j2` durchführbar. Bei den Orbitintegratoren `orbint_field` und `orbint_field2` dürfen jedoch wegen Punkt (ii) *nicht* die gegebenen zeitabhängigen Rotationsmatrizen nach den IERS-Konventionen verwendet werden, da ansonsten Präzession, Nutation und Polbewegung mitberücksichtigt würden, die Z-Achsen von erd- und raumfestem System also nicht zusammenfielen und folglich die Drehrate $\boldsymbol{\omega}$ ein zeitabhängiger, dreidimensionaler Vektor wäre. Deshalb werden für den Test dieser Integratoren die Rotationsmatrizen mithilfe der vereinfachten Beziehung $\boldsymbol{R}_\mathrm{E}^\mathrm{I}(t) = \boldsymbol{R}_3(\mathrm{GMST}_0 + \omega_\mathrm{E} \cdot t)$ berechnet (gleichförmige Drehrate ω_E). Ferner verwendet man ein Gravitationsfeldmodell, das neben dem Zentralterm nur aus dem zonalen Koeffizienten $\bar{C}_{2,0}$ besteht, sodass die aus `orbint_j2` und `orbint_field(2)` erhaltenen Orbits außerdem direkt miteinander verglichen werden können. Für den Vektor der Integrationszeiten wählt man bei diesem Test ein Zeitinkrement von einer Sekunde, d.h. `tspan` = `0:1:21600`.

Im Ergebnis bestehen alle Orbitintegratoren den Test auf die Erhaltung der Z-Komponente des Drehimpulsvektors, denn der zugehörige relative Fehler ist praktisch zu vernachlässigen, vgl. Abb. 4.13.

Die in Abbildung 4.14 dargestellten Differenzen zwischen den Komponenten der integrierten Bahnvektoren liegen auf einem akzeptablen Niveau von wenigen Nanometern (bei den Positionen) bzw. Pikometern pro Sekunde (bei den Geschwindigkeiten). Auch die Abweichungen zwischen den mit beiden Orbitintegratoren prädizierten Transitionsmatrizen steigen nur minimal an, s. Abb. 4.15 oben. Für den gewählten Testorbit sind kaum Unterschiede bei diesen Differenzen auszumachen, wenn man anstelle der herkömmlichen Berechnung des Marussi-Tensors die nicht-singuläre Methode nach Petrovskaya und Vershkov verwendet (Abb. 4.15 unten). Die Berechnungsdauer ist dabei indes länger (s. Tab. 4.5). Aus dem Test geht also hervor, dass die Lösung des Orbitintegrators `orbint_j2` derjenigen des Integrators `orbint_field` für $l_{\max} = 2$ mit hinreichender Genauigkeit entspricht. Die aufgezeigten Differenzen sind allein durch numerische Ungenauigkeiten bedingt und dadurch zu erklären, dass sich die Funktionen hinsichtlich der Berechnungsweise des DGL-Systems unterscheiden, das jeweils dem Matlab-DGL-Löser zugeführt wird (Kräftefunktion und Marussi-Tensor entsprechend den Formeln im dritten Kapitel).

Wenn man die Funktion `orbint_field2` anstelle `orbint_field` nutzt, liegen die in Abb. 4.14 und 4.15 gezeigten Differenzen zur Lösung der Funktion `orbint_j2` in denselben Größenordnungen (ohne Abb.). Dass die Rechenzeit bei diesem Integrator zwischen derjenigen von `orbint_j2` und `orbint_field` liegt (s. Tab. 4.5), lässt sich begreifen. Denn was den Rechenaufwand betrifft, stellt `orbint_field2` gerade den Mittelweg zwischen `orbint_j2` und `orbint_field` dar, weil der J_2-Marussi-Tensor wie in `orbint_j2` sozusagen direkt, die Beschleunigungen aber über den Gravitationsgradienten wie in `orbint_field` bestimmt werden.



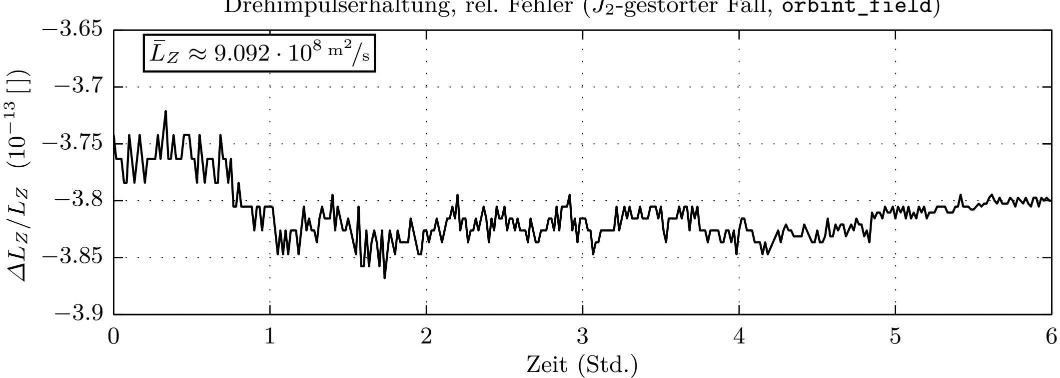


Abb. 4.13: Erhaltung der dritten Komponente des Drehimpulsvektors im J_2-gestörten Fall

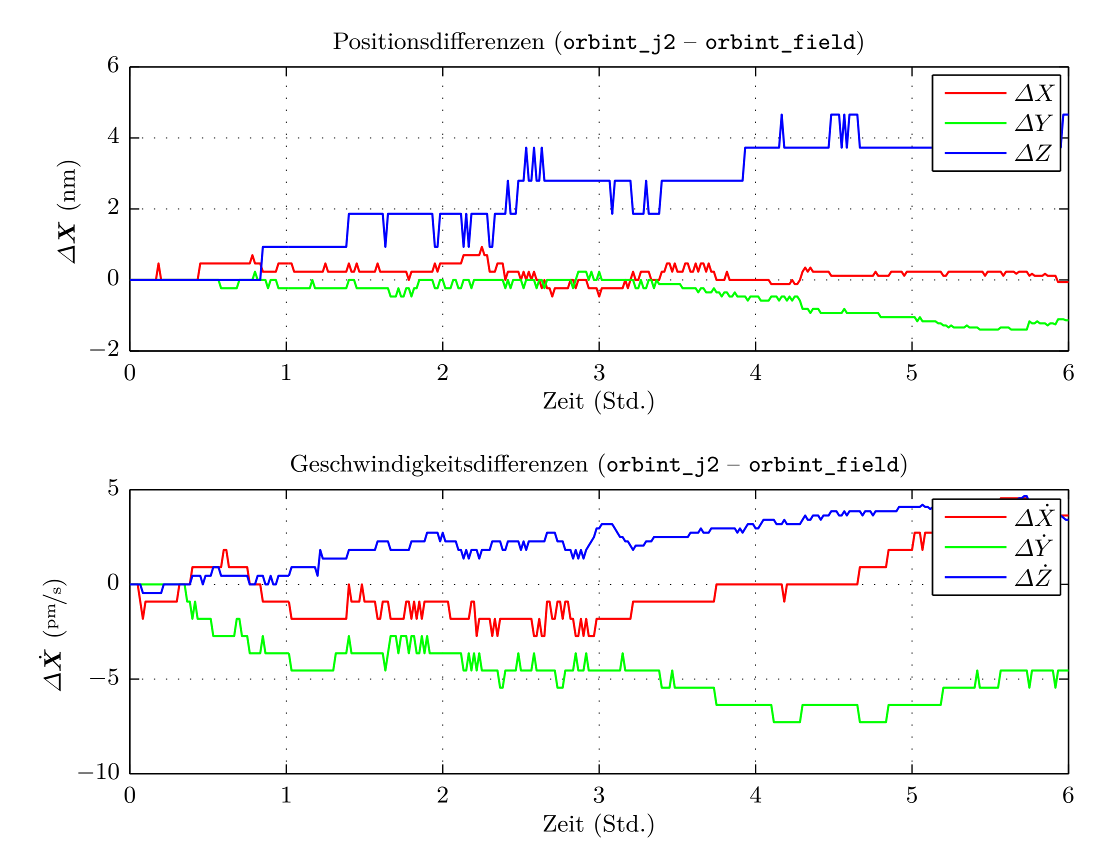


Abb. 4.14: `orbint_j2` vs. `orbint_field` im J_2-gestörten Fall (Bahnvektordifferenzen)

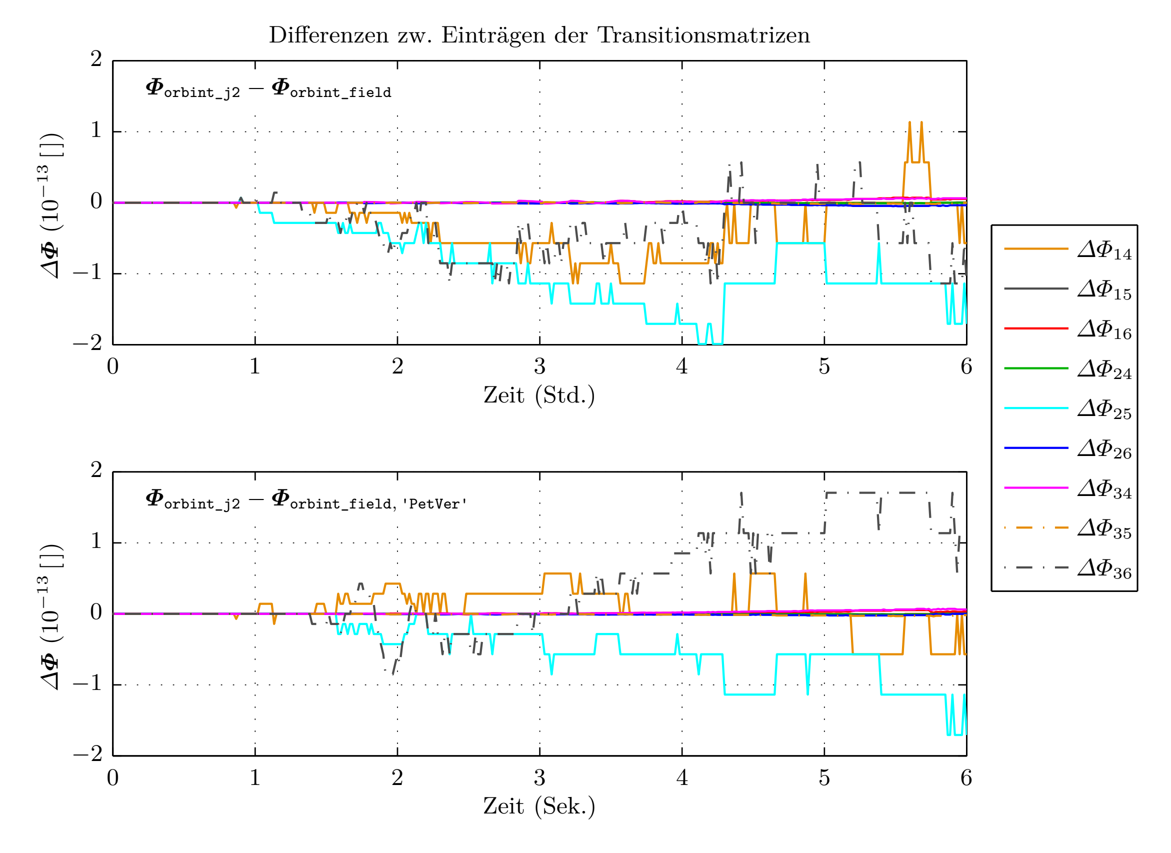


Abb. 4.15: `orbint_j2` vs. `orbint_field` im J_2-gestörten Fall (Differenzen zw. Transitionsmatrizen)

4.5.4 Vergleich mit unabhängigem Programm zur Integration von Satellitenorbits

Zur Kontrolle der Orbitintegratoren ist der Vergleich mit einem eigenständigen Bahnintegrationsprogramm sicherlich sinnvoll. Dadurch kann auch die Funktion `orbint_field` für ein vollständiges Gravitationsfeld höherer Ordnung überprüft werden.

Dem Buch von Montenbruck und Gill (2005) liegt eine Programmbibliothek in C++ (`SAT_Lib`) auf CD-ROM bei, die u.a. zur Bahnberechnung dient. Von Interesse ist hier die dort auf Seite 255 gestellte Aufgabe 7-1, deren Lösung man durch Ausführen des Quellcodes `ex71.cpp` erhält. Dabei werden Bahnintegrationen für drei Fälle (Referenz mit einem (10×10)-Gravitationsfeld, J_2-gestörter Fall und Keplerfall) durchgeführt und ebenfalls die zugehörigen Transitionsmatrizen berechnet. Der Programmcode `ex71.cpp` lässt sich leicht auf die hier gewählten Testbedingungen anpassen (Anfangswerte nach Gl. (4.8), Epoche 52640 MJD, Integrationszeit 6 Stunden). Desweiteren wurde im Programm die vorgesehene Normierung einiger Elemente der Transitionsmatrix deaktiviert. Das angepasste Programm liefert die in Abb. 4.16 gezeigten Ergebnisse, welche sich auf den letzten Zeitpunkt (6 Stunden bzw. 21600 s) beziehen.

```
Epoch (MJD) [d]      : 52640.0

GMST at epoch [rad] : 1.7494710664993

Time since epoch [s]: 21600.00


J10 State transition matrix (10x10 gravity field)

     -0.354707       0.619645      -2.854713    1259.025130  -2049.137618  -1885.794656
      0.097734       0.299512       1.137344    -967.221580    108.609630   1066.961619
    -24.602059      21.891799     -65.893092   43252.554443 -42009.280220 -31456.673483
      0.019690      -0.016565       0.052362     -33.538638     32.982239     24.991907
     -0.017899       0.016897      -0.050156      32.495932    -31.188766    -23.917544
     -0.002453       0.002211      -0.004413       3.173869     -3.069627     -1.306731

J2 State transition matrix

     -0.364691       0.627784      -2.880810    1275.726047  -2065.738252  -1898.278073
      0.106971       0.290724       1.163080    -984.106216    124.731399   1079.259254
    -24.599317      21.888593     -65.884898   43247.768151 -42004.436427 -31453.505631
      0.019685      -0.016560       0.052351     -33.531594     32.974500     24.987214
     -0.017895       0.016893      -0.050147      32.489520    -31.183004    -23.913934
     -0.002467       0.002224      -0.004451       3.198972     -3.093914     -1.325217

Keplerian State transition matrix

      0.828924      -0.350392       0.233438    -722.325697    -84.653542   -412.848717
     -0.949979       1.318655      -1.818477     964.246951  -1728.996370   -341.811991
    -24.700842      21.975344     -65.903158   43206.438380 -41966.014350 -31398.477506
      0.019878      -0.016760       0.052573     -33.624467     33.122191     24.990478
     -0.018040       0.016998      -0.050182      32.521229    -31.151326    -23.834010
     -0.000648       0.000605       0.000450      -0.000973      0.014367      1.014417


J10 State vector (10x10 gravity field)

   (X,  Y , Z ) =  -4946864.129332561  4735966.331423166   403209.774667544  [m]
   (Xp, Yp, Zp) =       -413.914749131       212.412506714   -7613.129365289  [m/s]

J2 State vector

   (X,  Y , Z ) =  -4947048.408116818  4736203.137417465   407061.890780893  [m]
   (Xp, Yp, Zp) =       -416.911064713       215.365522221   -7612.350540818  [m/s]

Keplerian state vector

   (X,  Y , Z ) =  -4959933.621637488  4731836.377951402   -51436.042619839  [m]
   (Xp, Yp, Zp) =        -54.275970947      -131.519263126   -7623.192619117  [m/s]
```

Abb. 4.16: Konsolen-Ausgabe des zum Vergleich herangezogenen C++-Programms

Für den Vergleich mit den eigenen Orbitintegratoren wurden diese Ergebnisse einfach von der Konsole in das Matlab-Programm (`orbint_test_main`) kopiert[1]. Die eigenen Matlab-Funktionen müssen zwecks des Vergleichs auf die gleichen Konstanten und das JGM3-Gravitationsfeldmodell, wie es in `SAT_Lib` definiert ist, Zugriff haben. Diese Größen werden gleich zu Beginn im Programm `orbint_test_main` festgelegt. Da in den vorherigen Tests der Kepler-Orbit und der J_2-gestörte Orbit also bereits unter Verwendung dieser Konstanten berechnet wurden, erfolgt am Ende des Programms `orbint_test_main` noch die Integration des bis zum Grad 10 gestörten Orbits (mittels

1 Der Vergleich ist also bei den Transitionsmatrizen auf sechs und bei den Zustandsvektoren auf neun Nachkommastellen begrenzt.

`orbint_field`). Danach werden die Zustandsvektoren sowie die Transitionsmatrizen der eigenen Orbitintegrationen von den in Abb. 4.16 aufgeführten Resultaten subtrahiert.

Im Keplerfall betragen die so berechneten Differenzen bei allen Einträgen der Transitionsmatrix Null, im J_2-gestörten Fall ist die größte Abweichung zwischen den Matrizen beim Eintrag Φ_{36} mit $1.3 \cdot 10^{-5}$ festzustellen und im J_{10}-gestörten Fall liegt der maximale Unterschied bei $\Delta\Phi_{34} = -0.000629$. Angesichts des Absolutbetrages der entsprechenden Matrixeinträge (s. Abb. 4.16) und der langen Integrationszeit sind diese Abweichungen akzeptabel.

In Tabelle 4.4 sind die Differenzen zwischen den Komponenten der Bahnvektoren aufgeführt. Diese sind zum einen durch numerische Ungenauigkeiten (Rundungsfehler etc.) und zum anderen durch die unterschiedlichen Verfahren zur numerischen Lösung der DGLen bedingt. Die etwas höheren Abweichungen beim J_{10}-Fall gegenüber dem J_2-Fall kommen möglicherweise dadurch zustande, dass in `SAT_Lib` die unnormierten Legendre-Funktionen in Verbindung mit den unnormierten Potentialkoeffizienten zur Bestimmung des Gravitationsgradienten verwendet werden (s. Funktion `AccelHarmonic`, Datei `SAT_Force.cpp`), was bekanntlich numerisch weniger robust ist.

Tab. 4.4: Vergleich der Zustandsvektoren (`SAT_Lib` vs. eigene Orbitintegrationen)

	Kepler-Fall	J_2-Fall	J_{10}-Fall
ΔX (m)	$2.21 \cdot 10^{-7}$	$-2.65 \cdot 10^{-7}$	$3.396 \cdot 10^{-6}$
ΔY (m)	$-2.06 \cdot 10^{-7}$	$1.97 \cdot 10^{-7}$	$-1.42 \cdot 10^{-6}$
ΔZ (m)	$4.5 \cdot 10^{-8}$	$-1.365 \cdot 10^{-6}$	$5.817 \cdot 10^{-5}$
$\Delta \dot{X}$ (m/s)	0	0	$-4.7 \cdot 10^{-8}$
$\Delta \dot{Y}$ (m/s)	0	0	$4.6 \cdot 10^{-8}$
$\Delta \dot{Z}$ (m/s)	0	0	$5 \cdot 10^{-9}$

Tab. 4.5: Rechenzeiten der Integratoren für 6-Stunden-Testorbits im J_2- und J_{10}-Fall

Integrator	Gravitationsfeld	$l_{\max}$	Tensormethode	Rechenzeit (Sek.)
`orbint_j2`	zonal	2	–	0.9
`orbint_field2`	zonal	2	–	1.9
`orbint_field`	zonal	2	herkömml./singulär	2.6
`orbint_field`	zonal	2	`'PetVer'`/nicht-singulär	6.1
`orbint_field`	vollst.	10	herkömml./singulär	4.9

Als Ergebnis des Tests kann festgestellt werden, dass die aufgezeigten Unterschiede zur Lösung des autonomen C++-Programms hinreichend klein sind, um von der korrekten Arbeitsweise der eigenen Orbitintegratoren auszugehen.

4.6 Test hinsichtlich der Fortpflanzung der Schätzfehlerkovarianzen

In diesem, ebenfalls im Programm `orbint_test_main` umgesetzten Test untersucht man, wie groß die durch die Linearisierung entstehenden Fehler bei der Fortschreibung der Kovarianzmatrix $\widehat{\boldsymbol{P}}_k$ sind. Dazu geht man in dieser Weise vor:

- Zuerst wählt man die gleichen Anfangsbedingungen aus wie schon bei den vorigen Tests, s. Gl. (4.8). Dieser Startvektor $\hat{\boldsymbol{s}}_{\text{ref}}$ ist zur Integration eines Referenzorbits bestimmt.
- Anschließend erzeugt man 100 variierte Startvektoren, indem man mittels der Matlab-Routine `randn` sechs mal 100 Realisierungen eines normalverteilten Rauschprozesses mit Mittelwert Null

und Standardabweichungen von 2 cm bzw. 0.1 mm/s erzeugt und mit den sechs Komponenten des Startvektors $\hat{\boldsymbol{s}}_{\mathrm{ref}}$ zusammenaddiert.

- Nun wird einerseits ein 600-Sekunden-Referenzorbit numerisch mithilfe `orbint_field` integriert, wobei das GGM02s-Feld, die zugehörigen Konstanten sowie `lmax` = 60 verwendet wird. Zum anderen werden in analoger Weise 100 Orbits integriert, ausgehend von den 100 variierten Startvektoren.
- Danach ermittelt man anhand der variierten Orbits und dem Referenzorbit bei 0, 60, und 600 Sekunden jeweils die sechs Standardabweichungen

$$\sigma_{\bar{s},\,\mathrm{ref}} = \left\{ \sigma_{\overline{X},\,\mathrm{ref}}\ ,\ \ldots\ ,\ \sigma_{\dot{\overline{Z}},\,\mathrm{ref}} \right\}$$

  und stellt mit den Fehlermaßen bei $t = 0\,\mathrm{s}$ die Schätzfehlerkovarianzmatrix $\widehat{\boldsymbol{P}}_{\mathrm{ref}}(t_0)$ auf. Diese Größen dienen hier als Referenz, weil die Bahnvektoren des Referenzorbits und der variierten Orbits nach dem *nicht-linearen* Systemmodell prädiziert wurden.
- Es folgt die lineare Fortschreibung der Matrix $\widehat{\boldsymbol{P}}_{\mathrm{ref}}(t_0)$ unter Verwendung der Transitionsmatrix basierend auf Gleichung (3.37) gemäß

$$\overline{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k \widehat{\boldsymbol{P}}_k \boldsymbol{\Phi}_k^{\top}$$

  mithilfe der Funktion `ekf_predict`, wobei die Integration ebenfalls mit `orbint_field`, dem obigen Gravitationsfeldmodell und `lmax2` = 60 für den Marussi-Tensor erfolgt. Das Systemrauschen entfällt hier ($\boldsymbol{Q} = \boldsymbol{0}$), da man allein die Genauigkeit der deterministischen Fehlerfortpflanzung untersuchen möchte. `ekf_predict` liefert die bis zum Zeitpunkt $t = 600\,\mathrm{s}$ prädizierten Kovarianzmatrizen als Zeitreihe `P_tsm` zurück[1].
- Schließlich extrahiert man die Standardabweichungen aus der Zeitreihe `P_tsm` an den Zeitpunkten $t = 60$ und 600 Sekunden und stellt diese den zuvor ermittelten Werten $\sigma_{\bar{s},\,\mathrm{ref}}$ gegenüber (s. Tabelle 4.6).

Tab. 4.6: Standardabweichungen beim Test der Schätzfehler-Fortpflanzung

Komponente	$\sigma_{\hat{s},\,\mathrm{ref}}$ @ $t = 0\,\mathrm{s}$	$\sigma_{\bar{s},\,\mathrm{ref}}$ @ $t = 60\,\mathrm{s}$	$\sigma_{\bar{s},\,\mathrm{tsm}}$ @ $t = 60\,\mathrm{s}$	$\sigma_{\bar{s},\,\mathrm{ref}}$ @ $t = 600\,\mathrm{s}$	$\sigma_{\bar{s},\,\mathrm{tsm}}$ @ $t = 600\,\mathrm{s}$
X (cm)	1.83	1.92	1.95	6.48	6.51
Y (cm)	2.11	2.16	2.19	5.71	5.80
Z (cm)	1.90	1.95	2.00	7.00	7.19
$\dot{X}$ (mm/s)	0.113	0.113	0.113	0.092	0.091
$\dot{Y}$ (mm/s)	0.099	0.099	0.099	0.081	0.080
$\dot{Z}$ (mm/s)	0.097	0.097	0.097	0.142	0.144

An den tabellierten Werten fällt zunächst ins Auge, dass sich die Schätzfehler der Geschwindigkeiten aufgrund der Systemdynamik innerhalb von 60 Sekunden praktisch nicht (bzw. vernachlässigbar wenig) verändern. Bei den Positionskomponenten übersteigt die Lösung mithilfe der Transitionsmatrix hier den entsprechenden Referenzwert um maximal 0.5 mm. Dies ist in dieser Anwendung akzeptabel, da der normalerweise noch hinzukommende Anteil des Systemrauschens bei der Prozessierung realer GRACE-Daten aus Gründen der Filterstabilität deutlich größer gewählt werden muss (im Bereich weniger Zentimeter), wie später noch deutlich wird. Nach 600 Sekunden unterscheiden sich $\sigma_{\bar{s},\,\mathrm{ref}}$ und $\sigma_{\bar{s},\,\mathrm{tsm}}$ bereits um maximal 1.9 mm (Z-Komponente), d.h. die Linearisierungsfehler

1 „tsm“ steht hierbei für „Transitionsmatrix“. Die hier mit „tsm“ indizierten Größen basieren auf der linearen Kovarianzprädiktion mithilfe der Transitionsmatrix.

steigen mit der Integrationszeit an. Im Falle anderer Orbitverdichtungen, die noch längere Integrationszeiten von etwa mehreren Stunden erfordern, würde somit das lineare Modell zur Fortschreibung der Schätzfehlerkovarianzen wahrscheinlich nicht mehr genügen.

Die Ergebnisse für $\sigma_{\bar{s},\,\mathrm{tsm}}$ verändern sich nur unwesentlich, wählt man für `lmax2` niedrigere Entwicklungsgrade. Mit `lmax2` = 10 (wie in Montenbruck und Gill, 2005, S. 255) erhält man bei 600 Sekunden hier Werte, die im Maximum nur 61 nm bzw. 454 pm/s von der angegebenen, bis Grad 60 berechneten Lösung $\sigma_{\bar{s},\,\mathrm{tsm}}$ abweichen. Mit `lmax2` = 2 betragen diese Differenzen 376 nm bzw. 782 pm/s. Bei 60 Sekunden sind die Differenzen mit etwa 2 nm bzw. 20 pm/s in beiden Fällen (`lmax2` = 10 od. 2) nochmals geringer, sodass es hier unbedenklich ist, nur die Potentialkoeffizienten bis Grad 2 oder 10 bei der Berechnung des Marussi-Tensors zu berücksichtigen. Zumindest bei derart kurzen Integrationszeiten wie hier wiegt also offenbar die Vernachlässigung der Terme höherer Ordnung bei der Linearisierung schwerer als der Effekt, der durch die Vernachlässigung der höheren Entwicklungsgrade in den Variationsgleichungen entsteht.

4.7 Test der EKF-Orbitverdichtung mithilfe simulierter Daten

Der nächste Schritt besteht nun darin, zu untersuchen, ob der gesamte Prozess der Orbitverdichtung mit den implementierten erweiterten Kalman-Filtern wirksam funktioniert, und insbesondere, ob sich damit ein simulierter GRACE-Orbit hinsichtlich der Genauigkeit verbessern lässt.

Die Tests der für die Orbitverdichtung verantwortlichen Funktionen `ekf_orbit` und `ekf_orbit2` erfolgen im Programm `ekf_test_main`, wobei man wie folgt vorgeht:

Schritt 1 – Berechnung der Referenzorbits für GRACE-A und -B

Um realen GRACE-Orbits nahe zu kommen, wurden als Anfangsbedingungen für die zwei zu bestimmenden Referenzorbits die entsprechenden Bahnvektoren aus dem erstem Eintrag der Datensätze `GraceA/52640.mat` bzw. `GraceB/52640.mat` gewählt. Mit diesen Startwerten soll hier jeweils ein 10-Minuten-Orbit mithilfe der Funktion `orbint_field` und unter Verwendung des Modells GGM02s bis Grad 20 integriert werden. Die benötigten Rotationsmatrizen berechnet man gemäß

$$\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}(t) = \boldsymbol{R}_3(\mathrm{GMST}_0 + \omega_{\mathrm{E}} \cdot t)$$

in der Funktion `rotation_ts` für alle Integrationszeitpunkte t. Der Stundenwinkel GMST_0 stammt dabei aus dem Datensatz `GraceX/52640.mat`. Die Rotationsmatrizen liegen dann als Zeitreihe im Intervall von einer Sekunde vor (`Ri2e1s`). Somit können die Orbitintegrationen erfolgen und man erhält schließlich die simulierten, zehnminütigen Referenzorbits für GRACE-A und -B.

Schritt 2 – Erzeugung der Navigationslösungen und K-Band-Beobachtungen

Um die Navigationslösungen der Satelliten nachzubilden, reduziert man zuerst das Sampling der Zeitreihen des Referenzorbits auf 60 Sekunden und addiert dann auf jede Komponente einzeln mittels der `randn`-Routine normalverteiltes, mittelwertfreies Rauschen mit einer Standardabweichung von 0.02 m bei den Positions- und 10^{-4} m/s bei den Geschwindigkeitskomponenten. In den so erzeugten Datensätzen der Navigationslösungen (`navposA`, `navvelA`, `navposB`, `navvelB`) werden noch die positiven Quadratwurzeln der Kovarianzen benötigt[1]. Diese werden für beide Satelliten und alle Zeitpunkte identisch angenommen zu:

$$\sqrt{\sigma_{XX}} = \sqrt{\sigma_{YY}} = \sqrt{\sigma_{ZZ}} = 0.02\,\mathrm{m}, \quad \sqrt{\sigma_{XY}} = \sqrt{\sigma_{XZ}} = \sqrt{\sigma_{YZ}} = 0\,\mathrm{m}$$
$$\sqrt{\sigma_{\dot{X}\dot{X}}} = \sqrt{\sigma_{\dot{Y}\dot{Y}}} = \sqrt{\sigma_{\dot{Z}\dot{Z}}} = 10^{-4}\,\mathrm{m/s}, \quad \sqrt{\sigma_{\dot{X}\dot{Y}}} = \sqrt{\sigma_{\dot{X}\dot{Z}}} = \sqrt{\sigma_{\dot{Y}\dot{Z}}} = 0\,\mathrm{m/s}$$

1 D.h. jeweils sechs Spalten entsprechend der in Abschn. 4.1.1 dargelegten Struktur.

Setzt man von Null verschiedene Kreuzkorrelationen an, wie sie in den realen Navigationslösungen vorkommen, hat dies keine signifikante Auswirkung auf die späteren Ergebnisse, weshalb diese hier entfallen können. Die verrauschten K-Band-Beobachtungen werden generiert, indem man zunächst aus den Referenzorbits mithilfe der Funktion `grace_parms` die K-Band-Referenz-Zeitreihen ($\rho,\ \dot{\rho}$) berechnet, danach das Sampling auf 5 Sekunden reduziert und normalverteiltes Rauschen mit einer Standardabweichung von $\sigma_\rho = 10\,\mu\text{m}$ bzw. $\sigma_{\dot{\rho}} = 1\,\mu\text{m/s}$ addiert.

Schritt 3 – Prozessierung der simulierten Daten aus Schritt 2 durch die EKF-Funktionen und anschließender Vergleich mit den Referenzwerten

Zur Verdichtung der simulierten Navigationslösungen aus Schritt 2 testet man nun die beiden Funktionen `ekf_orbit` und `ekf_orbit2` mit unterschiedlichen Orbitintegratoren und variiert z.T. noch andere Parameter, siehe Tabelle 4.7. Aufgrund der zahlreichen Einstellungsmöglichkeiten könnte man noch weitere Tests durchführen; hier beschränkt man sich auf sechs Tests, anhand derer die Auswirkungen der Einstellungen auf das Ergebnis der Orbitverdichtung demonstriert werden. In Tabelle 4.7 sind nur diejenigen Aspekte aufgeführt, durch die sich die Tests voneinander unterscheiden. An die Hauptfunktionen sind zudem zwei weitere wichtige Stellgrößen zu übergeben, die für alle Tests – soweit in Tab. 4.7 nicht anders angegeben – wie folgt gewählt wurden:

(i) Matrix `Q`, die das Systemrauschen modelliert und entsprechend Abschnitt 3.2.4.2 aus den Variablen `qpos` und `qvel` berechnet wird. Hier legen wir die Werte mit `qpos` $= 0.001\,\text{m}$ und `qvel` $= 10^{-5}\,\text{m/s}$ fest. Beim Ausprobieren mit anderen Werten hat sich gezeigt, dass mit größeren Werten der Vergleich zu den Referenzorbits eher schlechter ausfällt und andererseits mit kleineren Werten die Abweichungen zur Referenz kaum geringer sind.

(ii) Parameter `cov_sel`, der angibt, nach welcher Methode die Matrix `Q` an die prädizierten Kovarianzen angebracht wird. Beim Vergleich mit der Referenz sind keine wesentlichen Unterschiede festzustellen, wenn man die Methode ändert, d.h. das Ergebnis ist weitgehend unabhängig von der Wahl des Parameters `cov_sel`. Da sich aber bei der Methode `'ric'` die vergleichsweise kürzeste Rechenzeit ergibt, wurde stets dieses Verfahren verwendet.

Tab. 4.7: Übersicht der verschiedenen Tests zur EKF-Orbitverdichtung

Test Nr.	Hauptfunktion	Nutzung K-Band	Orbitintegrator `lmax(/lmax2)`	sonstige Parameter	Rechenzeit (Sek.)
1	`ekf_orbit`	✗	`orbint_field` 20/20	—	3.6
2	`ekf_orbit`	✗	`orbint_field2` 20	—	2.1
3	`ekf_orbit`	✗	`orbint_field2` 10	—	1.6
4	`ekf_orbit2`	✓	`orbint_field2` 20	$\sigma_{\dot{\rho}} = 1\,\mu\text{m/s}$ $\sigma_{0,\texttt{navvel}} = 1$	15
5	`ekf_orbit2`	✓	`orbint_field2` 20	$\sigma_{\dot{\rho}} = 0.1\,\mu\text{m/s}$ $\sigma_{0,\texttt{navvel}} = 1$	15
6	`ekf_orbit2`	✓	`orbint_field2` 20	$\sigma_{\dot{\rho}} = 1\,\mu\text{m/s}$, `qvel` $= 10^{-4}\,\text{m/s}$ $\sigma_{0,\texttt{navvel}} = 0.5$	15

Nach der Prozessierung mit `ekf_orbit` bzw. `ekf_orbit2` liegen die verdichteten Orbits jeweils in zeitlicher Auflösung von einer Sekunde vor und können mit den Referenzbahnen verglichen werden.

Um festzustellen, ob die jeweilige getestete Kalman-Filterung einen Genauigkeitsgewinn einbringt, wertet man zunächst die Navigationslösungen mit Bezug auf die Referenzorbits statistisch aus. Dies geschieht hier, indem man das Minimum, Maximum und die Standardabweichung der Differenzen

zwischen den Komponenten des Referenzorbits und den Komponenten der simulierten Navigationslösung berechnet. Die ermittelten Standardabweichungen liegen in derselben Größenordnung wie das aufmodulierte Rauschen (s. Tab. 4.8a und 4.8b). Auf Seite 94 und 95 sind außerdem sämtliche Ergebnisse der getesteten Orbitverdichtungen links für GRACE-A und rechts für GRACE-B tabellarisch verzeichnet. Zu beachten ist, dass man für den Vergleich wie zuvor nur diejenigen Zustandsvektoren der Orbits herangezogen hat, die 60 Sekunden auseinander liegen.

Vergleicht man die Ergebnisse der ersten Orbitverdichtung (Tab. 4.8c bzw. 4.8d) mit der Ausgangssituation (Tab. 4.8a bzw. 4.8b), erkennt man, dass sich sämtliche Standardabweichungen reduziert haben. Durch die Kalman-Filterung haben sich die Genauigkeiten also – im Gesamten betrachtet – tatsächlich verbessert. Wie Abbildung 4.17 und 4.18 anhand der Bahnvektor-Differenzen der GRACE-A-Orbits veranschaulicht, liegen manche Komponenten der Navigationslösung aber *vereinzelt* näher an der Referenz als die betreffende durch das Kalman-Filter korrigierte Komponente.

Tab. 4.8: Vergleich der simulierten Navigationslösungen und der verdichteten Orbits mit den Referenzorbits

(a): Vergleich zwischen Referenzorbit und simulierter Navigationslösung (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{ref} - X_{nav}$ (m)	−0.035	0.027	0.019
$Y_{ref} - Y_{nav}$ (m)	−0.033	0.054	0.025
$Z_{ref} - Z_{nav}$ (m)	−0.047	0.025	0.023
$\dot{X}_{ref} - \dot{X}_{nav}$ (m/s)	−0.000098	0.000125	0.000078
$\dot{Y}_{ref} - \dot{Y}_{nav}$ (m/s)	−0.000134	0.000263	0.000110
$\dot{Z}_{ref} - \dot{Z}_{nav}$ (m/s)	−0.000166	0.000239	0.000108

(b): Vergleich zwischen Referenzorbit und simulierter Navigationslösung (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{ref} - X_{nav}$ (m)	−0.009	0.044	0.024
$Y_{ref} - Y_{nav}$ (m)	−0.059	0.004	0.023
$Z_{ref} - Z_{nav}$ (m)	−0.036	0.025	0.021
$\dot{X}_{ref} - \dot{X}_{nav}$ (m/s)	−0.000163	0.000121	0.000101
$\dot{Y}_{ref} - \dot{Y}_{nav}$ (m/s)	−0.000195	0.000208	0.000105
$\dot{Z}_{ref} - \dot{Z}_{nav}$ (m/s)	−0.000152	0.000090	0.000095

(c): Vergleich zwischen Referenzorbit und nach Test 1 verdichtetem Orbit (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{ref} - X_{EKF}$ (m)	−0.004	0.018	0.011
$Y_{ref} - Y_{EKF}$ (m)	−0.019	0.013	0.009
$Z_{ref} - Z_{EKF}$ (m)	−0.023	0.016	0.015
$\dot{X}_{ref} - \dot{X}_{EKF}$ (m/s)	−0.000016	0.000094	0.000046
$\dot{Y}_{ref} - \dot{Y}_{EKF}$ (m/s)	−0.000135	0.000014	0.000057
$\dot{Z}_{ref} - \dot{Z}_{EKF}$ (m/s)	−0.000095	−0.000003	0.000055

(d): Vergleich zwischen Referenzorbit und nach Test 1 verdichtetem Orbit (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{ref} - X_{EKF}$ (m)	0.003	0.030	0.018
$Y_{ref} - Y_{EKF}$ (m)	−0.024	0.016	0.011
$Z_{ref} - Z_{EKF}$ (m)	−0.031	0.002	0.014
$\dot{X}_{ref} - \dot{X}_{EKF}$ (m/s)	−0.000046	0.000045	0.000029
$\dot{Y}_{ref} - \dot{Y}_{EKF}$ (m/s)	0.000013	0.000209	0.000087
$\dot{Z}_{ref} - \dot{Z}_{EKF}$ (m/s)	−0.000128	0.000005	0.000060

(e): Vergleich zwischen Referenzorbit und nach Test 2 verdichtetem Orbit (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{ref} - X_{EKF}$ (m)	−0.004	0.018	0.011
$Y_{ref} - Y_{EKF}$ (m)	−0.019	0.013	0.009
$Z_{ref} - Z_{EKF}$ (m)	−0.023	0.016	0.015
$\dot{X}_{ref} - \dot{X}_{EKF}$ (m/s)	−0.000016	0.000094	0.000046
$\dot{Y}_{ref} - \dot{Y}_{EKF}$ (m/s)	−0.000135	0.000014	0.000057
$\dot{Z}_{ref} - \dot{Z}_{EKF}$ (m/s)	−0.000095	−0.000003	0.000055

(f): Vergleich zwischen Referenzorbit und nach Test 2 verdichtetem Orbit (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{ref} - X_{EKF}$ (m)	0.003	0.030	0.018
$Y_{ref} - Y_{EKF}$ (m)	−0.024	0.016	0.011
$Z_{ref} - Z_{EKF}$ (m)	−0.031	0.002	0.014
$\dot{X}_{ref} - \dot{X}_{EKF}$ (m/s)	−0.000046	0.000045	0.000029
$\dot{Y}_{ref} - \dot{Y}_{EKF}$ (m/s)	0.000013	0.000209	0.000087
$\dot{Z}_{ref} - \dot{Z}_{EKF}$ (m/s)	−0.000128	0.000005	0.000060

Tab. 4.8: Vergleich der simulierten Navigationslösungen und der verdichteten Orbits mit den Referenzorbits

(g): Vergleich zwischen Referenzorbit und nach Test 3 verdichtetem Orbit (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	−0.011	0.122	0.054
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.218	0.016	0.130
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.113	0.408	0.131
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.001047	0.001980	0.001003
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	−0.002478	0.000726	0.001408
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.002554	0.005002	0.002464

(h): Vergleich zwischen Referenzorbit und nach Test 3 verdichtetem Orbit (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	−0.005	0.147	0.064
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.237	0.045	0.135
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.178	0.350	0.122
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000989	0.002045	0.000961
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	−0.002402	0.000983	0.001389
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.003023	0.004526	0.002324

(i): Vergleich zwischen Referenzorbit und nach Test 4 verdichtetem Orbit (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	−0.009	0.011	0.007
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.015	0.015	0.009
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.025	0.013	0.017
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000042	0.000018	0.000020
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	−0.000056	0.000035	0.000025
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.000072	0.000050	0.000051

(j): Vergleich zwischen Referenzorbit und nach Test 4 verdichtetem Orbit (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	0.006	0.033	0.020
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.025	0.010	0.013
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.027	−0.002	0.014
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000042	0.000070	0.000040
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	0.000013	0.000131	0.000060
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.000187	−0.000010	0.000076

(k): Vergleich zwischen Referenzorbit und nach Test 5 verdichtetem Orbit (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	−0.009	0.011	0.007
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.015	0.015	0.009
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.025	0.013	0.017
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000042	0.000018	0.000020
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	−0.000056	0.000035	0.000025
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.000072	0.000050	0.000051

(l): Vergleich zwischen Referenzorbit und nach Test 5 verdichtetem Orbit (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	0.006	0.033	0.020
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.025	0.010	0.013
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.027	−0.002	0.014
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000042	0.000070	0.000040
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	0.000013	0.000131	0.000060
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.000187	−0.000010	0.000076

(m): Vergleich zwischen Referenzorbit und nach Test 6 verdichtetem Orbit (GRACE A)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	−0.015	0.019	0.010
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.021	0.017	0.011
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.031	0.018	0.016
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000073	0.000088	0.000048
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	−0.000068	0.000089	0.000050
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.000085	0.000094	0.000062

(n): Vergleich zwischen Referenzorbit und nach Test 6 verdichtetem Orbit (GRACE B)

	Minimum	Maximum	Standard-abweichung
$X_{\mathrm{ref}} - X_{\mathrm{EKF}}$ (m)	0.003	0.032	0.019
$Y_{\mathrm{ref}} - Y_{\mathrm{EKF}}$ (m)	−0.026	0.011	0.016
$Z_{\mathrm{ref}} - Z_{\mathrm{EKF}}$ (m)	−0.033	0.011	0.013
$\dot{X}_{\mathrm{ref}} - \dot{X}_{\mathrm{EKF}}$ (m/s)	−0.000080	0.000066	0.000040
$\dot{Y}_{\mathrm{ref}} - \dot{Y}_{\mathrm{EKF}}$ (m/s)	−0.000042	0.000128	0.000045
$\dot{Z}_{\mathrm{ref}} - \dot{Z}_{\mathrm{EKF}}$ (m/s)	−0.000191	0.000063	0.000075

Obwohl man beim zweiten Test den Integrator `orbint_field2` nutzt, dessen Fehlerfortpflanzung auf dem vereinfachten J_2-Kräftemodell basiert, fällt der Vergleich mit der Referenzbahn nicht schlechter als im ersten Test aus (s. Tab. 4.8e u. 4.8f). Wie nach dem vorherigen Test in Abschnitt 4.6 zu erwarten war, bleibt die Genauigkeit der prädizierten Zustandsvektoren annähernd erhalten. Hier betragen die Unterschiede zum ersten Test bei den Standardabweichungen der Positionen nur etwa

10^{-9} m und bei den Geschwindigkeiten etwa 10^{-11} m/s. Dieses Testresultat bestätigt also ebenso, dass die nach dem vereinfachten, linearen Modell prädizierten Kovarianzen – zumindest im Fall der hier verwendeten Orbits – genügend genau sind, um noch einen zuverlässigen Korrekturschritt zu ermöglichen.

Verringert man aber darüberhinaus auch noch die Prädiktionsgenauigkeit des Systemzustands, indem man z.B. den maximalen Entwicklungsgrad 10 wählt (Test 3), fällt der Vergleich der berechneten Orbits mit den Referenzorbits deutlich schlechter aus (s. Tab. 4.8g/4.8h). Werden in diesem Fall die Werte für `qpos` und `qvel` angehoben, ergeben sich höchstens Abweichungen zur Referenz wie bei der Navigationslösung, geringere Differenzen lassen sich nicht erreichen. Dieser Test zeigt also, dass sich durch die Kalman-Filterung die Genauigkeit des resultierenden Orbits nur dann erhöhen kann, wenn man zur Prädiktion – zumindest zur Prädiktion der Zustandsvektoren – dasselbe Kräftemodell nutzt, wie es dem gegebenen, verrauschten Orbit zugrunde liegt.

Im vierten Test werden nun die simulierten K-Band-Relativgeschwindigkeiten in die Orbitbestimmung miteinbezogen (Funktion `ekf_orbit2`). Dabei bleiben zwar die Abweichungen der Bahnvektoren zu den Referenzwerten etwa auf dem gleichen Genauigkeitsniveau wie im zweiten Test (vgl. Tab. 4.8i/4.8j mit Tab. 4.8e/4.8f). Allerdings passen die verdichteten Orbits nun erheblich besser zu den K-Band-Messungen $\dot{\rho}$. Der Vergleich mit den K-Band-Daten erfolgt in Abb. 4.19 und 4.20. Dabei wurden zum einen aus den beiden Navigationslösungen und zum anderen aus den paarweise verdichteten GRACE-Bahnen die K-Band-Größen ρ und $\dot{\rho}$ mithilfe der Funktion `grace_parms` ermittelt und die Abweichungen zu den simulierten K-Band-Beobachtungen (5-Sek.-Sampling) über der Zeit dargestellt. Wie aus Abb. 4.20 hervorgeht, liegen die Differenzen zwischen den Relativgeschwindigkeiten im vierten Test bei nur 10^{-8} m/s. Offensichtlich lässt sich also die *relative* Genauigkeit der erhaltenen Satellitenbahnen *zueinander* deutlich steigern, wenn man die K-Band-Messungen als Beobachtungsgrößen in die EKF-Orbitverdichtung mitaufnimmt. Die Differenzen der K-Band-Relativentfernungen (Abb. 4.20) weisen jedoch in allen Fällen etwa dieselbe Größenordnung auf, weil diese hier nicht in die EKF-Orbitverdichtung eingehen.

Mit den beiden letzten Tests werden zwei verschiedene Ansätze dafür präsentiert, wie man die Satellitenbahnen noch stärker an die K-Band-Beobachtungen $\dot{\rho}$ anpassen kann. Die erste Idee ist, die festzulegende Standardabweichung $\sigma_{\dot{\rho}}$ genauer anzunehmen als sie eigentlich ist. So erhalten wir mit $\sigma_{\dot{\rho}} := 0.1$ µm/s im fünften Test nochmals reduzierte Differenzen $\Delta\dot{\rho}$ um 10^{-10} m/s (s. Abb. 4.20). Der zweite Ansatz besteht darin, den Parameter `qvel` zu erhöhen und die Geschwindigkeiten der Navigationslösung mittels $\sigma_{0,\texttt{navvel}}$ herabzugewichten, was ein ähnliches Ergebnis liefert. In beiden Fällen bewirken die Maßnahmen, dass die K-Band-Beobachtung $\dot{\rho}$ beim Korrekturschritt stärker berücksichtigt wird. Je überspitzter die genannten Parameter $\sigma_{\dot{\rho}}$, `qvel` und $\sigma_{0,\texttt{navvel}}$ gewählt werden, desto eher kommt der Korrekturschritt durch die K-Band-Messungen einer Ausgleichung unter Zwang gleich, wobei die Zwangsbedingung die jeweilige K-Band-Beobachtung $\dot{\rho}$ darstellt. Auf die Standardabweichungen zum Referenzorbit haben diese Maßnahmen nur geringen Einfluss, vgl. Tab. 4.8i mit Tab. 4.8k u. 4.8m.

Natürlich kommt es vor allem auf den weiteren Verwendungszweck der verdichteten Satellitenbahnen an, ob eine gegenüber dem Standardfall (Test 4) verstärkte Anpassung an die K-Band-Relativgeschwindigkeiten nötig ist. Im Anwendungsfall der SGG-Auswertung, bei der die K-Band-Messungen mit der aus den Bahnvektoren berechneten Größe $\|\dot{\boldsymbol{X}}_{12}\|^2$ kombiniert werden (Gl. (2.13)), könnte es sich eventuell als nützlich erweisen, diese Anpassung bis zu einem gewissen Ausmaß anzuheben. Die oben genannten Parameter können daher vom Benutzer der Toolbox im Programmfester (Abb. 4.2) individuell eingestellt werden.

Abschließend kann man im Hinblick auf die durchgeführten Tests feststellen, dass die zur Orbitverdichtung implementierten Kalman-Filter-Mechanismen wie gewünscht arbeiten, da sich das Rauschen der simulierten Navigationslösungen mindern bzw. deren Orbitgenauigkeit nachweislich verbessern ließ. Somit gehen wir nun zur Verarbeitung realer GRACE-Daten über.

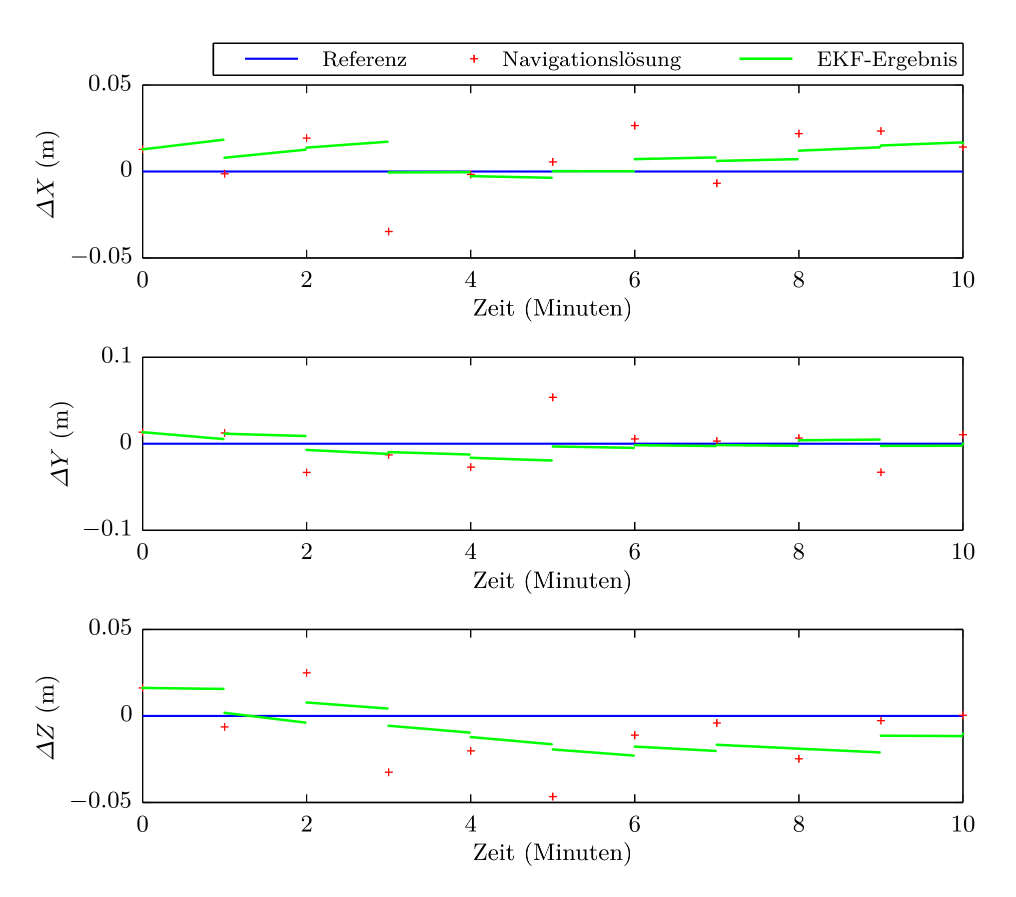


Abb. 4.17: Verdichteter Orbit vs. Referenzorbit (Positionen, GRACE A, Test 1)

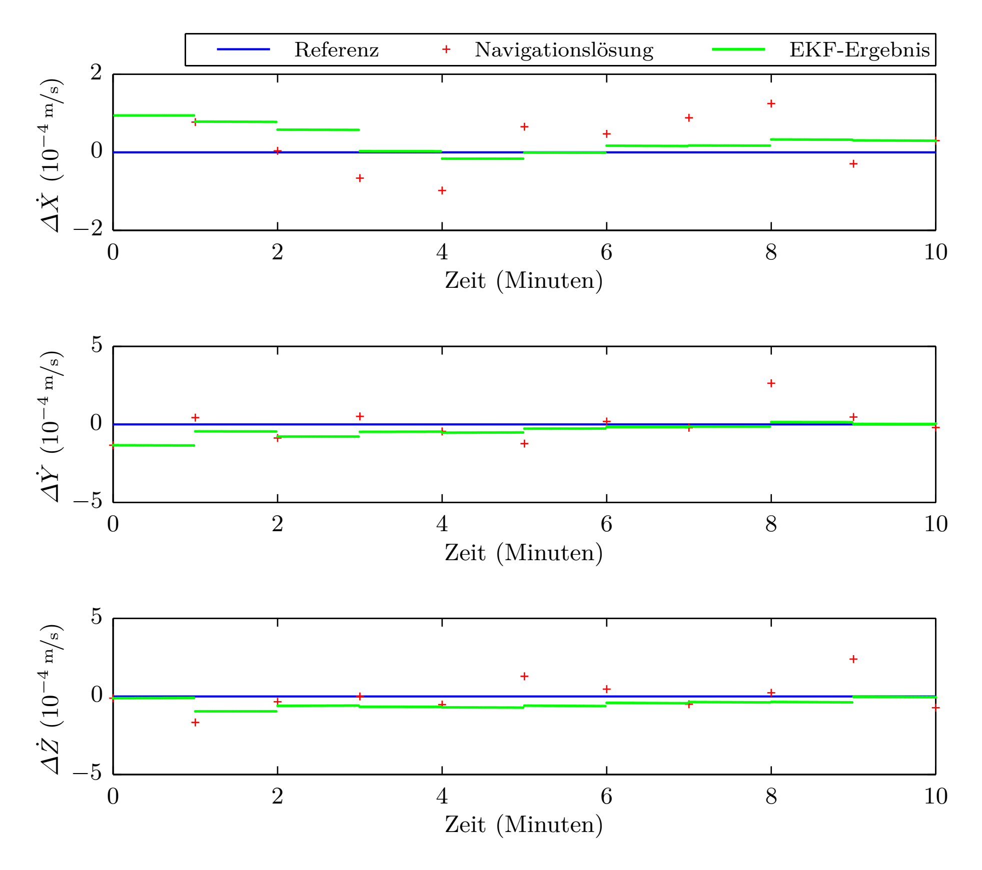


Abb. 4.18: Verdichteter Orbit vs. Referenzorbit (Geschwindigkeiten, GRACE A, Test 1)

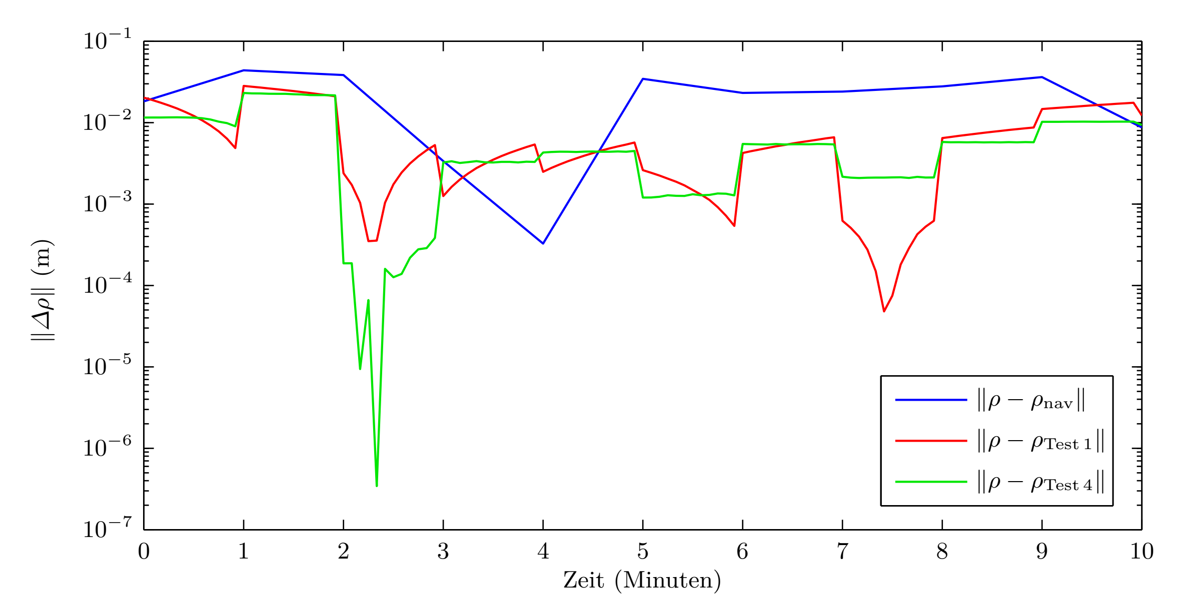


Abb. 4.19: Vergleich der K-Band-Relativentfernungen

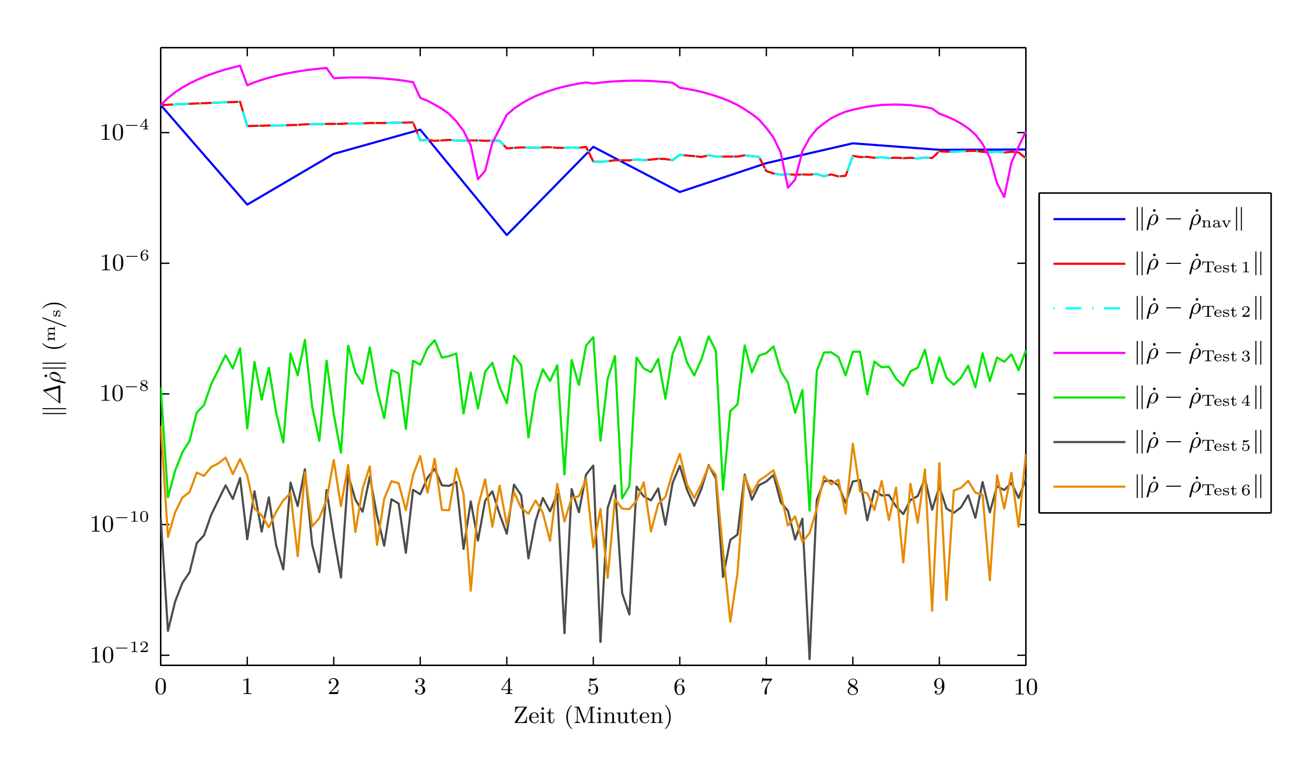


Abb. 4.20: Vergleich der K-Band-Relativgeschwindigkeiten

5 Kapitel 5 Prozessierung realer GRACE-Daten

5.1 Anmerkungen zu den verwendeten realen GRACE-Daten

Unter „realen GRACE-Daten“ seien hier diejenigen Daten verstanden, die der Toolbox in den Verzeichnissen `\GraceA`, `\GraceB` und `\GraceX` beiliegen. Wie bereits in Abschnitt 4.1.1 erwähnt, beinhalten die in den Ordnern `\GraceA` und `\GraceB` gespeicherten `mat`-Dateien u.a. die Navigationslösungen des Produkts GNV1B („GPS Navigation Data Format Record“*). Die Datengrundlage der im Verzeichnis `\GraceX` vorliegenden Dateien ist das sogenannte „K-Band Ranging Data Product“* KBR1B. Die `mat`-Dateien wurden für diese Arbeit folgendermaßen erstellt: zuerst wurden beide Level-1B-Produkte (GNV1B und KBR1B) über das PO.DAAC[1] bezogen. Generell sollte man dabei unbedingt darauf achten, die aktuellste Version der GNV1B-Navigationslösung (hier: „release 01“ aus dem Jahr 2005) zu wählen. Danach hat mein Betreuer die Daten in das nun vorliegende Matlab-kompatible Format konvertiert und zudem die Navigationslösungen, die ursprünglich nur bezüglich des erdfesten Systems gegeben waren, in das raumfeste Bezugssystem transformiert. Die dafür verwendeten Rotationsmatrizen wurden nach den IERS-Konventionen, also unter Berücksichtigung von Präzession, Nutation und Polbewegung, bestimmt und sind ebenfalls in den Dateien enthalten (Variablen `Re2i` und `Ri2e`).

Aus dem dritten Test in Abschnitt 4.7 ist klar geworden, dass man zur Verdichtung eines gegebenen, nicht-kinematischen Orbits mithilfe des implementierten erweiterten Kalman-Filters möglichst dasselbe dynamische Modell ansetzen sollte, welches dem gegebenen Orbit zugrunde liegt, um eine verbesserte Genauigkeit der resultierenden Bahnvektoren gegenüber den gegebenen Bahnvektoren zu erhalten (Reduzierung des Messrauschens). Leider sind nähere Informationen darüber, wie die am JPL produzierten GNV1B-Orbits aus den kinematischen GPS-Beobachtungen der GRACE-Satelliten gewonnen werden, in den üblichen GRACE-Dokumenten[2] nicht zu finden.

Laut dem verantwortlichen Mitarbeiter am JPL, Herrn Dr. William I. Bertiger, handelt es sich bei den GNV1B-Orbits um *reduziert-dynamische*, die einer stochastischen Orbitbestimmung entstammen. Bei der Kräftefunktion nutze man derzeit u.a. folgende Modelle:

- Gravitationsfeldmodell GGM02C (bis einschließlich Grad/Ordnung 200)
- Atmosphärenreibung nach DTM94[3]
- Globales Ozeangezeitenmodell basierend auf FES2004[4]

Hinzu kämen außerdem Modelle für Sonnenstrahlungsdruck und Erdalbedo. Die Akzelerometer-Daten würden dabei jedoch nicht verwendet.

Da die Implementierung all dieser Modelle im Rahmen dieser Arbeit zu aufwendig wäre und auch weitere Details, wie etwa die Bestimmung der Anfangsbedingungen der GNV1B-Orbits, nicht klar

∗ Bezeichnungen aus Case et al. (2010).
1 http://podaac.jpl.nasa.gov/gravity/grace
2 Dazu zählen z.B. Case et al. (2010), Bettadpur (2007), Wu et al. (2006).
3 „Drag temperature model“ 94, siehe Berger et al. (1998).
4 „Finite element solution“ 2004, siehe Lyard et al. (2006).

sind, kann hier zwangsläufig keine Reduzierung des Messrauschens der Navigationslösungen bzw. keine gesteigerte Genauigkeit der verdichteten Orbits gegenüber der Navigationslösung hinsichtlich der Komponenten des Bahnvektors erzielt werden. Allerdings können mithilfe der Toolbox die Orbits auch mit den umgesetzten einfacheren Kräftemodellen derart verdichtet werden, dass sie sehr gut mit den hochgenau gemessenen K-Band-Relativgeschwindigkeiten harmonieren – und zwar besser im Vergleich zu den Navigationslösungen. Dies wurde bereits bei der Verdichtung simulierter Orbits anhand der K-Band-Vergleiche in Abschnitt 4.7 belegt und trifft auch im Falle der Verdichtung realer Orbits zu, wie sich im Folgenden noch zeigen wird (Abschn. 5.3.5).

5.2 Vorab-Analyse der GRACE-Daten

Um sicherzugehen, dass sich die realen GRACE-Daten zur Orbitverdichtung eignen, d.h. die Navigationslösungen und die K-Band-Messungen insbesondere keine großen Fehler/Ausreißer aufweisen, sollten sie im Vorfeld der Verdichtung zumindest durch einige einfache Tests überprüft werden. Im hierfür erstellten Programm `grace_analysis_main` werden aus den Navigationslösungen K-Band-Größen berechnet, die dann mit den tatsächlichen K-Band-Messungen verglichen werden. Das Programm sucht zudem nach Datenlücken, die gelegentlich bei den K-Band-Beobachtungen vorkommen können und meldet eventuelle Fundstellen im MATLAB-Kommandofenster.

Positionen und Geschwindigkeiten vs. K-Band

In der Funktion `grace_parms` sind die Beziehungen

$$\rho = \langle \boldsymbol{X}_2 - \boldsymbol{X}_1, \boldsymbol{e}_{12} \rangle \quad \text{und} \quad \dot{\rho} = \langle \dot{\boldsymbol{X}}_2 - \dot{\boldsymbol{X}}_1, \boldsymbol{e}_{12} \rangle \quad \text{mit} \quad \boldsymbol{e}_{12} = \frac{\boldsymbol{X}_2 - \boldsymbol{X}_1}{\|\boldsymbol{X}_2 - \boldsymbol{X}_1\|}$$

gemäß Gl. (2.1) bis (2.3) implementiert. Damit liefert `grace_parms` aus den raumfesten Navigationslösungen der beiden Satelliten die K-Band-Größen ρ_{nav} und $\dot{\rho}_{\text{nav}}$. Diese subtrahiert man von den echten K-Band-Messungen, die nun als Referenz dienen sollen und erhält die Abweichungen $\Delta\rho$, $\Delta\dot{\rho}$ als Zeitreihe mit der zeitlichen Auflösung von 60 Sekunden (s. Abb. 5.1). In Tabelle 5.1 sind die entsprechenden Minima, Maxima und Standardabweichungen eingetragen. Vergleicht man die hier berechneten Standardabweichungen $\sigma_{\Delta\rho}$ bzw. $\sigma_{\Delta\dot{\rho}}$ mit den Mittelwerten der in den Datensätzen `navpos_i` bzw. `navvel_i` bereits vorliegenden Fehlermaßen (vgl. Tab. 2.1), stellt man fest, dass sie in derselben Größenordnung liegen (Positionskomponenten um 2 cm, Geschwindigkeitskomp. um 0.02 mm/s). Diesen Umstand kann man als Indiz dafür werten, dass die gegebenen Fehlermaße der Navigationslösungen durchaus realistisch sind.

Zufolge BERTIGER ET AL. (2010, Abschn. 3.1) wird wohl seit 1. Mai 2009 bei der Produzierung der GNV1B-Daten ein neuer Ansatz verwendet, bei dem die Phasenmehrdeutigkeiten für einzelne GPS-Empfänger getrennt gelöst werden. Die Autoren konnten damit die Standardabweichungen zwischen den K-Band-Relativentfernungen und der aus den GPS-Beobachtungen ermittelten Entfernung ($\widehat{=}\ \rho_{\text{nav}}$) von durchschnittlich 14.3 mm auf 4.1 mm verbessern. Die hier berechnete Standardabweichung von ca. 13.4 mm (Tab. 5.1) entspricht also den Erwartungen, da dieser GNV1B-Datensatz im Jahr 2005 erstellt wurde.

Tab. 5.1: Analyse der realen GRACE-Daten

	Minimum	Maximum	Standard-abweichung
$\Delta\rho$ (m)	-0.033909544	0.043333370	0.013361674
$\Delta\dot{\rho}$ (m/s)	$-7.55870 \cdot 10^{-5}$	$6.50815 \cdot 10^{-5}$	$1.89306 \cdot 10^{-5}$
$\Delta\ddot{\rho}$ (m/s²)	$-1.04503 \cdot 10^{-5}$	$1.09111 \cdot 10^{-5}$	$1.74356 \cdot 10^{-6}$
Δf (m/s²)	$-6.52125 \cdot 10^{-6}$	$7.85531 \cdot 10^{-6}$	$1.01056 \cdot 10^{-6}$

Beschleunigungen vs. K-Band

Um zu verifizieren, dass auch die beobachteten realen K-Band-Relativbeschleunigungen $\ddot{\rho}$ zu den Navigationslösungen passen, sollen sie einer aus den Navigationslösungen berechneten Entsprechung $\ddot{\rho}_{\text{nav}}$ gegenübergestellt werden. Hierfür eignet sich die Beziehung

$$\ddot{\rho}_{\text{nav}} = \langle \ddot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle + \frac{1}{\rho_{\text{nav}}} \left(\|\dot{\boldsymbol{X}}_{12}\|^2 - \dot{\rho}_{\text{nav}}^2 \right)$$

nach Gleichung (2.6). Darin ergeben sich die Beschleunigungen $\ddot{\boldsymbol{X}}_{12} = \ddot{\boldsymbol{X}}_2 - \ddot{\boldsymbol{X}}_1$, indem man die Geschwindigkeiten $\dot{\boldsymbol{X}}_1$ und $\dot{\boldsymbol{X}}_2$ numerisch differenziert. Dies wurde hier mithilfe der Funktion `firstderiv` bewerkstelligt, da die MATLAB-Funktion `gradient` das Rauschen weitaus mehr verstärkt. `firstderiv` nutzt einen Differentiationsfilter n-ter Ordnung (Funktion `cdtaylor`). Dabei handelt es sich um die Taylor-Approximation des zentralen Differenzenquotienten bis zu einer wählbaren Ordnung n, wobei im Prinzip der ideale Verlauf des Differentiators im Frequenzraum (Winkelhalbierende) umso besser angenähert wird, je höher man den Entwicklungsgrad bzw. die Ordnung der Taylor-Reihe wählt. Hier verbessern sich bereits ab Ordnung 4 die Resultate beim Vergleich mit den K-Band-Beobachtungen nicht mehr, siehe Abb. 5.2. Folglich müssten die restlichen Differenzen $\Delta\ddot{\rho} = \ddot{\rho} - \ddot{\rho}_{\text{nav}}$ hauptsächlich durch die Ungenauigkeiten der Geschwindigkeiten $\dot{\boldsymbol{X}}_1$ und $\dot{\boldsymbol{X}}_2$ bzw. der K-Band-Messungen $\ddot{\rho}$ bedingt sein.

Gemessene vs. a priori Kräftefunktion

Schließlich kann man noch einen Vergleich durchführen zwischen der Kräftefunktion, die sich aus den gegebenen realen Daten ergibt, und derjenigen Kräftefunktion, die man aus einem a priori Gravitationsfeldmodell (hier EGM96) berechnet.

Nach Gleichung (2.7) gilt der Zusammenhang

$$\langle \ddot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle = \ddot{\rho} - \frac{\|\dot{\boldsymbol{X}}_{12}\|^2 - \dot{\rho}^2}{\rho}.$$

Den Ausdruck $\langle \ddot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle$ auf der linken Seite, der gerade die Kräftefunktion f in LOS-Richtung darstellt, kann man demzufolge durch

$$f_{\text{gem.}} = \ddot{\rho} - \frac{1}{\rho_{\text{nav}}} \left(\langle \dot{\boldsymbol{X}}_{12}, \dot{\boldsymbol{X}}_{12} \rangle - \dot{\rho}^2 \right)$$

ausschließlich aus den gegebenen realen Daten berechnen. Andererseits liefert die Funktion `gradpshs` unter Verwendung des EGM96-Modells aus den erdfesten, sphärischen Satellitenpositionen nach den entsprechenden Transformationen mithilfe der Rotationsmatrizen `Ri2eA` bzw. `Ri2eB` die raumfesten Beschleunigungen $\ddot{\boldsymbol{X}}_1$ bzw. $\ddot{\boldsymbol{X}}_2$. Daraus erhält man die a priori Kräftefunktion zu

$$f_{\text{field}} = \langle \ddot{\boldsymbol{X}}_2 - \ddot{\boldsymbol{X}}_1, \boldsymbol{e}_{12} \rangle$$

und anschließend die in Abbildung 5.2 gezeigten Differenzen $\Delta f = f_{\text{gem.}} - f_{\text{field}}$, die durchweg sehr klein sind. Wenn man anstatt des EGM96-Modells das neuere EGM08 verwendet, reduzieren sich die Differenzen Δf um etwa eine Größenordnung. Die berechnete Standardabweichung verbessert sich dabei von $\sigma_{\Delta f} \doteq 10^{-6}\,\mathrm{m/s^2}$ (Tabelle 5.1) auf ca. $3.13 \cdot 10^{-7}\,\mathrm{m/s^2}$. Insofern scheinen die vorliegenden Daten also keine groben Fehler aufzuweisen und eignen sich völlig zur weiteren Verarbeitung – der Orbitverdichtung mit dem Hauptprogramm `ekf_orbit_main`.

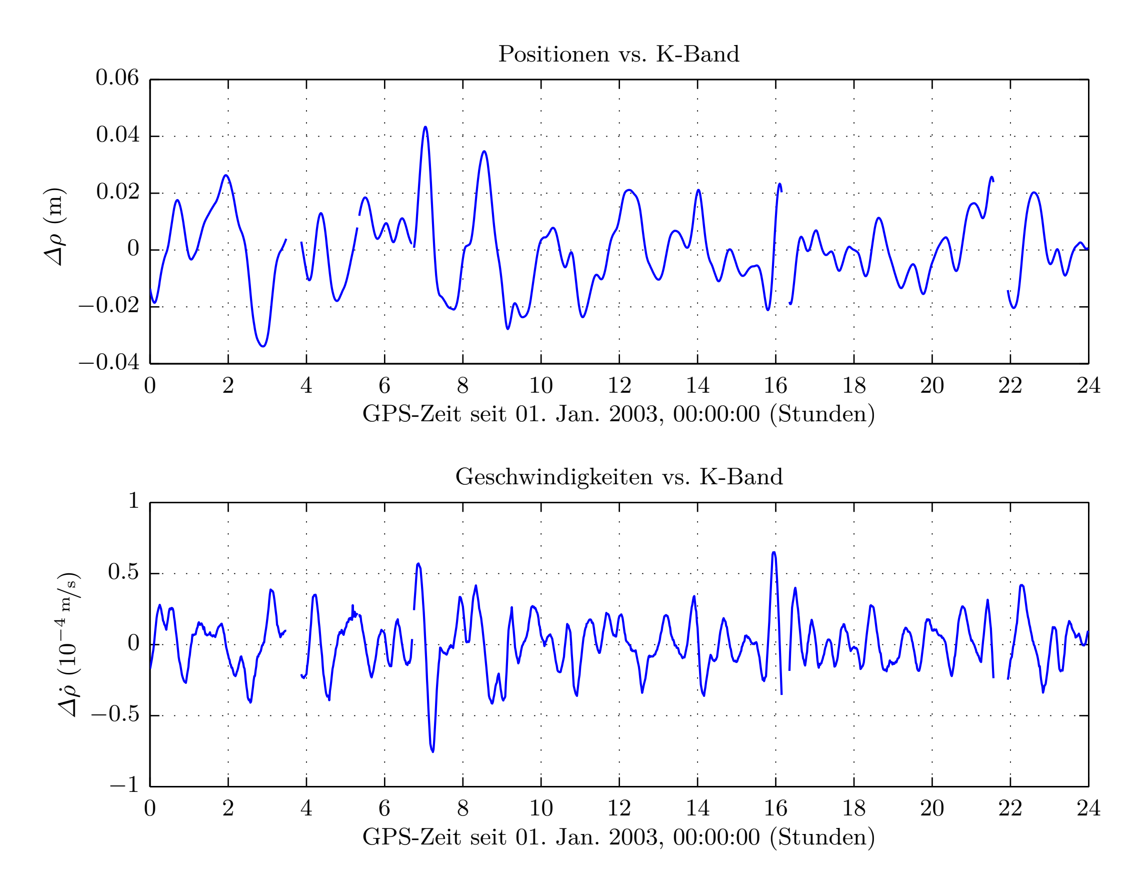


Abb. 5.1: Analyse der realen GRACE-Daten (Differenzen $\Delta\rho$, $\Delta\dot{\rho}$)

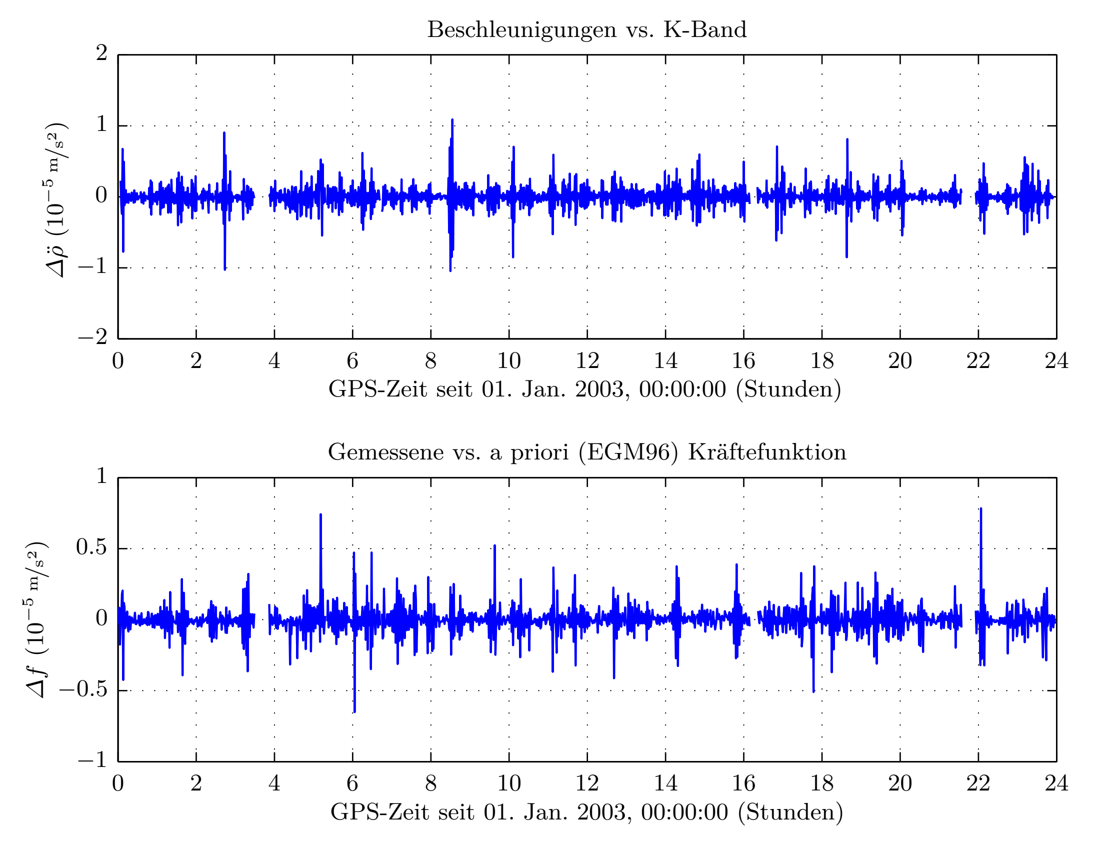


Abb. 5.2: Analyse der realen GRACE-Daten (Differenzen $\Delta\ddot{\rho}$, Δf)

5.3 Verdichtung der realen GRACE-Orbits

Wie schon aus Abschnitt 4.1 hervorgeht, ist in der Toolbox das Hauptprogramm `ekf_orbit_main` dafür zuständig, reale GRACE-Orbits der gegebenen Form zu verdichten. Der Aufbau und die zwei Programmfenster zum Einlese- und Verarbeitungsteil wurden dort bereits beschrieben. Im Ausgabeteil des Hauptprogramms kann man über ein drittes GUI aus zwanzig durchnummerierten Grafiken wählen, die zur Visualisierung der Ergebnisse dienen. Tabelle 5.2 gibt Auskunft über die Zuordnung zwischen der Grafiknummer und dem jeweiligen Inhalt der Grafik.

Tab. 5.2: Übersicht der verschiedenen Ausgabegrafiken des Hauptprogramms `ekf_orbit_main`

Grafik Nr.	Präfix des Namens der Ausgabedatei	Beschreibung
1/2	`fig01/fig02`	bei Messungsupdates geschätzte Verbesserungen $\boldsymbol{\delta\hat{s}}_k$ der prädizierten Zustandsvektoren sowie Differenzen zwischen den korrigierten Zustandsvektoren und der Navigationslösung (Positions-/Geschwindigkeitskomponenten, GRACE A)
3/4	`fig03/fig04`	wie oben nur für GRACE B
5/6	`fig05/fig06`	Standardabweichungen (der Positionen/Geschw.) beider verdichteten Orbits
7	`fig07/fig07b`	K-Band-Vergleich: Differenzen zwischen K-Band-Beobachtungen ρ, $\dot{\rho}$ und den entsprechenden Größen, die aus den Navigationslösungen bzw. den verdichteten Orbits berechnet wurden
8	`fig08`	Darstellung des GRACE-A-Orbits auf einer Weltkarte in Rektangularprojektion (quadratische Plattkarte)
9/10	`fig09/fig10`	Vergleich zwischen verdichtetem GRACE-A-Orbit und der Navigationslösung auf der Plattkarte: Norm des Differenzvektors $\widehat{\boldsymbol{X}} - \boldsymbol{X}_{\mathrm{nav}}$ (`fig09`) bzw. des Vektors $\dot{\widehat{\boldsymbol{X}}} - \dot{\boldsymbol{X}}_{\mathrm{nav}}$ (`fig10`)
11/12	`fig11/fig12`	Darstellung der geschätzten Korrekturen (GRACE A) auf der Plattkarte: Norm des Differenzvektors $\widehat{\boldsymbol{X}} - \overline{\boldsymbol{X}}$ (`fig11`) bzw. des Vektors $\dot{\widehat{\boldsymbol{X}}} - \dot{\overline{\boldsymbol{X}}}$ (`fig12`)
13/14	`fig13/fig14`	Vergleich zwischen verdichtetem GRACE-B-Orbit und der Navigationslösung auf der Plattkarte: Norm des Differenzvektors $\widehat{\boldsymbol{X}} - \boldsymbol{X}_{\mathrm{nav}}$ (`fig13`) bzw. des Vektors $\dot{\widehat{\boldsymbol{X}}} - \dot{\boldsymbol{X}}_{\mathrm{nav}}$ (`fig14`)
15/16	`fig15/fig16`	Darstellung der geschätzten Korrekturen (GRACE B) auf der Plattkarte: Norm des Differenzvektors $\widehat{\boldsymbol{X}} - \overline{\boldsymbol{X}}$ (`fig15`) bzw. des Vektors $\dot{\widehat{\boldsymbol{X}}} - \dot{\overline{\boldsymbol{X}}}$ (`fig16`)
17/18	`fig17/fig18`	Vergleich zwischen Navigationslösung und K-Band auf der Plattkarte: Betrag der Differenz $\Delta\rho$ (`fig17`) bzw. $\Delta\dot{\rho}$ (`fig18`)
19/20	`fig19/fig20`	Vergleich zwischen verdichtetem Orbit und K-Band auf der Plattkarte: Betrag der Differenz $\Delta\rho$ (`fig19`) bzw. $\Delta\dot{\rho}$ (`fig20`)

Hier wurden zur Orbitverdichtung exemplarisch die Datensätze gewählt, welche die Beobachtungen vom 1. Januar 2003 verzeichnen (`52640.mat`). Für diese Orbits führte man mehrere Prozessierungen mit unterschiedlichen Einstellungen durch, siehe Tab. 5.3. Nachfolgend werden die Ergebnisse dargelegt, verglichen und interpretiert.

5.3.1 Verarbeitung 1: kurzer Bahnabschnitt mit den bereits in den Simulationen verwendeten Prozessierungsparametern

Um die grundsätzliche Wirkungsweise einiger Einstellparameter zu verdeutlichen, wurden zunächst nur kurze Bahnstücke bis 900 Sekunden (15 Minuten) verdichtet. Abbildung 5.8 zeigt die Bodenspur dieses Bahnabschnitts von GRACE-A, deren Verlauf charakteristisch für polnahe Bahnen ist. Bei

Tab. 5.3: Optionen und Rechenzeiten der durchgeführten Prozessierungen

	Verarbeitung 1	Verarbeitung 2	Verarbeitung 3	Verarbeitung 4	Verarbeitung 5
Stop bei (Sek.)	900	900	900	86340	86340
Orbitintegrator	`orbint_field2`	`orbint_field2`	`orbint_field2`	`orbint_field2`	`orbint_j2`
Gravitationsfeld-modell/`lmax`	GGM02s/60	GGM02s/60	GGM02s/60	GGM02s/60	GGM02s/(2)
Kovarianzmethode	Riccati (`'ric'`)	Riccati (`'ric'`)	Riccati (`'ric'`)	Riccati (`'ric'`)	Riccati (`'ric'`)
`qpos` (cm)	1	2.5	2.5	2.5	6
`qvel` (mm/s)	0.01	0.1	0.25	0.25	0.6
Nutzung K-Band	✓	✓	✓	✓	✓
$\sigma_{\dot{\rho}}$ (µm/s)	1	0.7	0.4	0.4	0.4
$\sigma_{0,\texttt{navvel}}$	1	0.8	0.2	0.2	0.2
Rechenzeit (HH:MM:SS)	00:00:58	00:00:59	00:00:59	01:25:36	00:05:24

der ersten Prozessierung wählte man zunächst einfach dieselben Werte für die Parameter `qpos`, `qvel`, $\sigma_{\dot{\rho}}$ und $\sigma_{0,\texttt{navvel}}$ wie im vierten Test in Abschnitt 4.7 bei der Prozessierung simulierter Orbits.

Anhand Abb. 5.3 und Abb. 5.4 (Ergebnisgrafiken `fig01` und `fig02`) ist ersichtlich, dass die bei den Messungsupdates geschätzen Verbesserungen des prädizierten Zustandsvektors relativ klein bleiben (rote Markierungen), während die Differenzen zwischen den korrigierten Zustandsvektoren und der Navigationslösung mit der Zeit eher ansteigen (blaue Kreuze). Dieses Anstiegsverhalten ist aber hier nicht erwünscht. Zufolge der Anmerkungen in Abschnitt 5.1 sind im dynamischen Modell, das bei der Produzierung der GNV1B-Daten genutzt wird, weitaus mehr Störbeschleunigungen mitberücksichtigt als im eigenen, beim Prädiktionsschritt verwendeten Kräftemodell. Die Situation ist hier also eine andere als noch bei den Tests in Abschnitt 4.7, als nämlich die Simulation der Navigationslösung und die Prädiktion der Systemzustände unter Verwendung ein und desselben Kräftemodells erfolgte. Vielmehr muss man in diesem Fall (bei der Prozessierung realer Daten) den Navigationslösungen aufgrund des erweiterten Kräftemodells ein potentiell höheres Genauigkeitsniveau zubilligen als den prädizierten Zustandsvektoren. Offenbar ‚vertraut' aber das Filter in dieser ersten Verarbeitung den prädizierten Systemzuständen noch zu sehr, worauf auch die anfangs deutlich abfallenden Schätzfehlervarianzen (Abb. 5.5 und 5.6) hindeuten. Folglich macht es Sinn, bei der folgenden, zweiten Verarbeitung die Parameter des Systemrauschens, `qpos` und `qvel`, anzuheben. Damit die K-Band-Messungen $\dot{\rho}$ bei der Korrektur mehr Gewicht bekommen und mithin die Abweichungen $\Delta\dot{\rho}$ noch weiter reduziert werden, welche hier bereits meist unterhalb von 0.1 µm/s liegen (s. Abb. 5.7 und Tab. 5.4), verringert man im nächsten Durchlauf auch die Werte der Parameter $\sigma_{\dot{\rho}}$ und $\sigma_{0,\texttt{navvel}}$ ein wenig.

Tab. 5.4: K-Band-Vergleich zur ersten Verarbeitung

(a) Navigationslösungen vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.002597682	0.005025348	0.002341338
$\Delta\dot{\rho}$	(m/s)	$-1.70208 \cdot 10^{-5}$	$2.80200 \cdot 10^{-5}$	$1.75543 \cdot 10^{-5}$

(b) Berechnete Orbits vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.006137524	0.015474921	0.005052320
$\Delta\dot{\rho}$	(m/s)	$-1.51938 \cdot 10^{-7}$	$7.96568 \cdot 10^{-8}$	$3.81263 \cdot 10^{-8}$

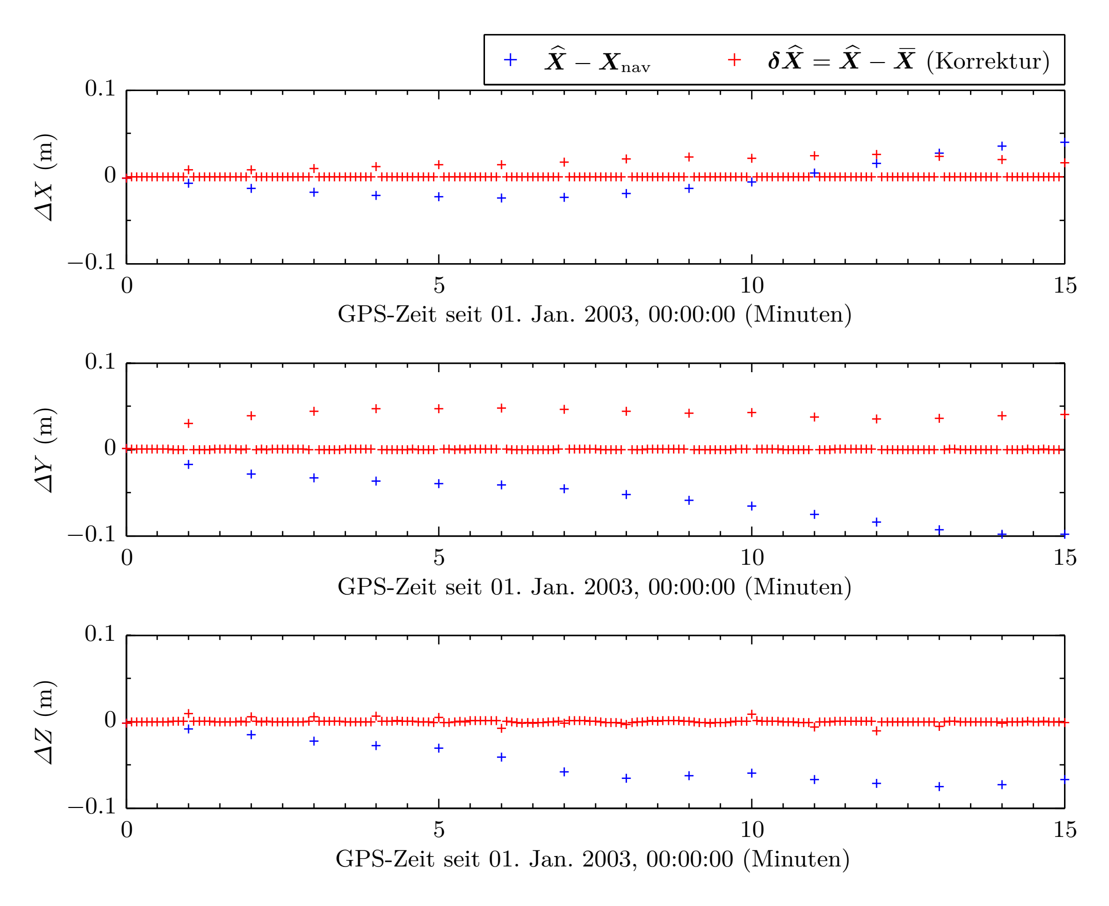


Abb. 5.3: `fig01` – geschätzte Korrekturen und Vergleich zur Navigationslösung (Positionen, GRACE A)

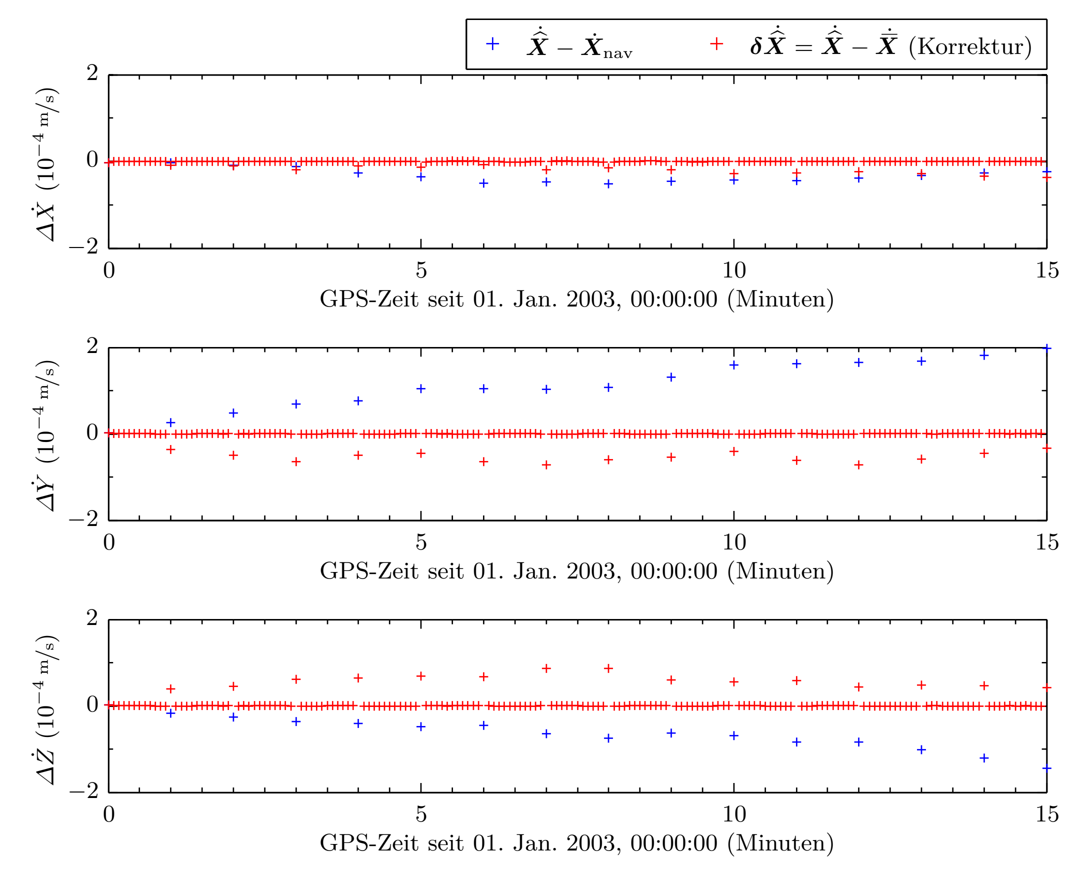


Abb. 5.4: `fig02` – geschätzte Korrekturen u. Vergl. zur Navigationslösung (Geschwindigkeiten, GRACE A)

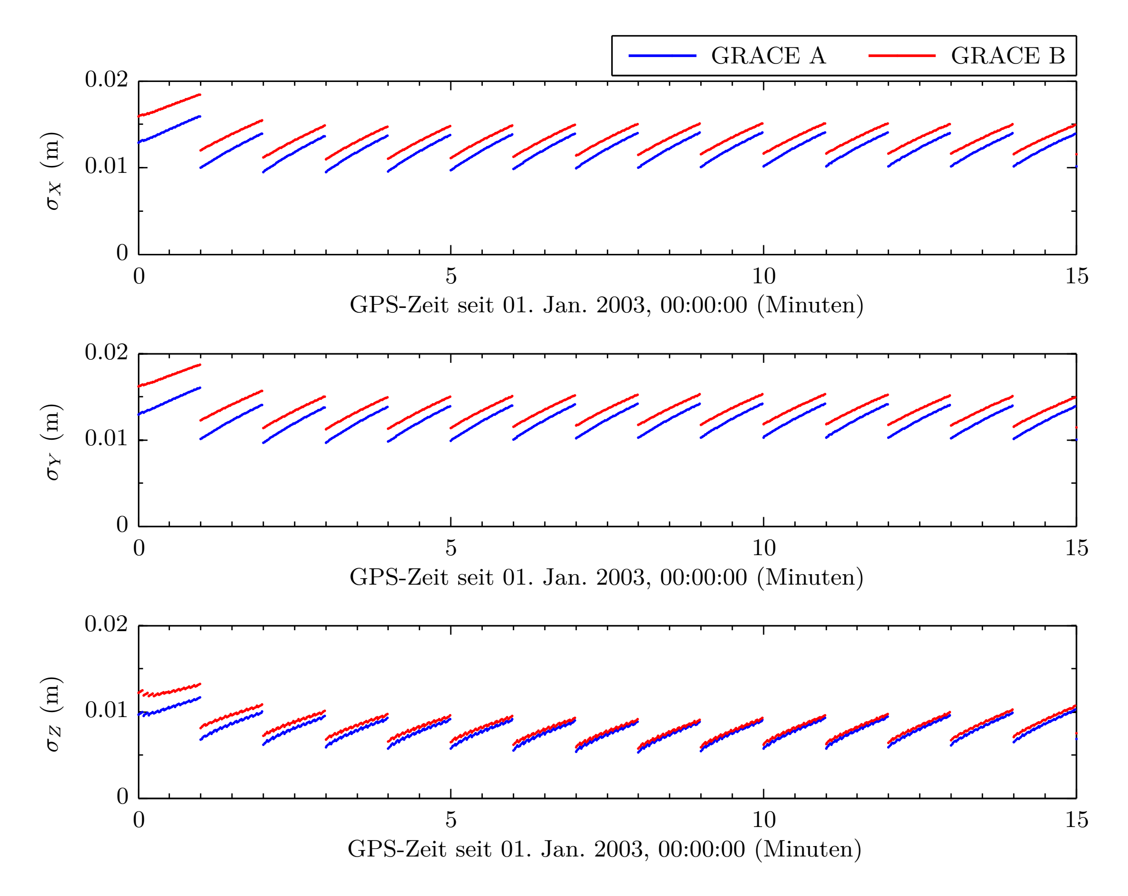


Abb. 5.5: `fig05` – Standardabweichungen der Positionskomponenten (GRACE A & B)

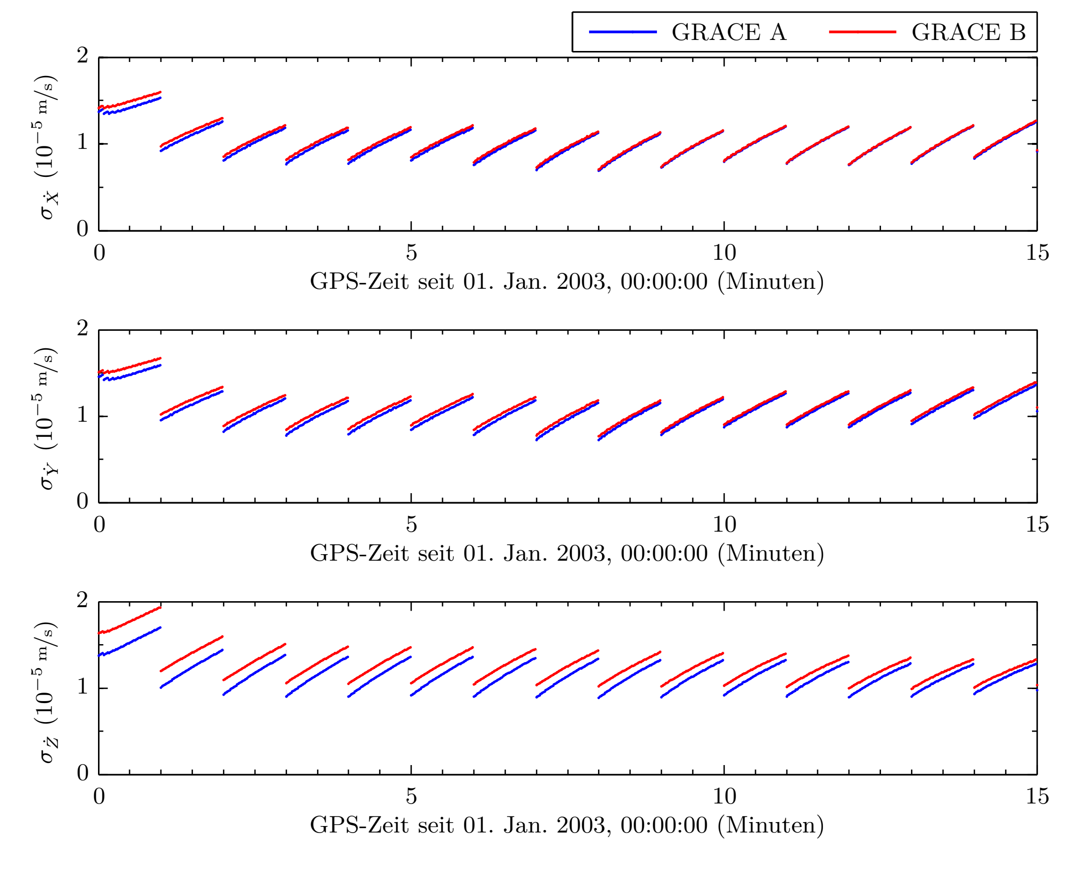


Abb. 5.6: `fig06` – Standardabweichungen der Geschwindigkeitskomponenten (GRACE A & B)

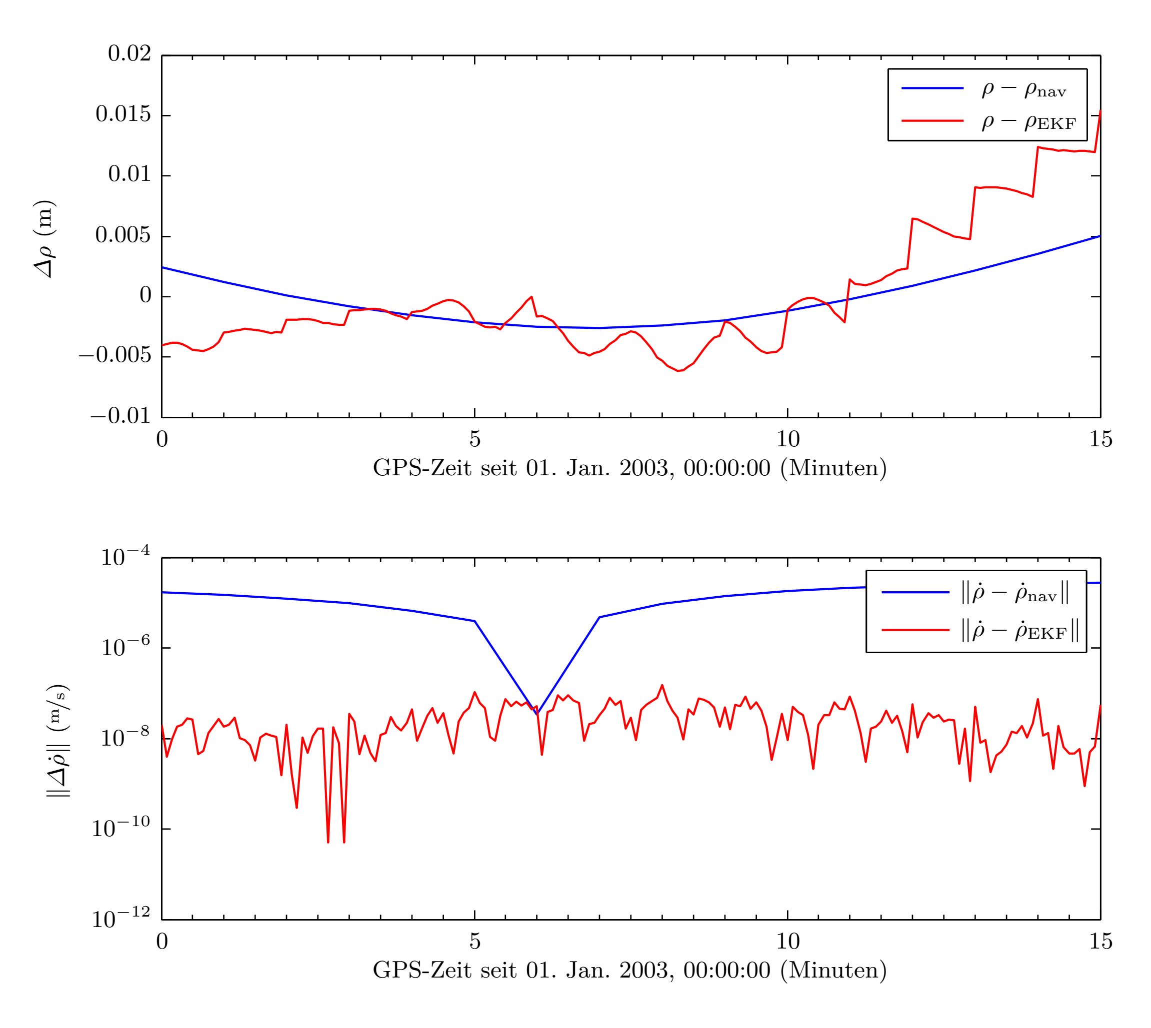


Abb. 5.7: `fig07b` – K-Band-Vergleich

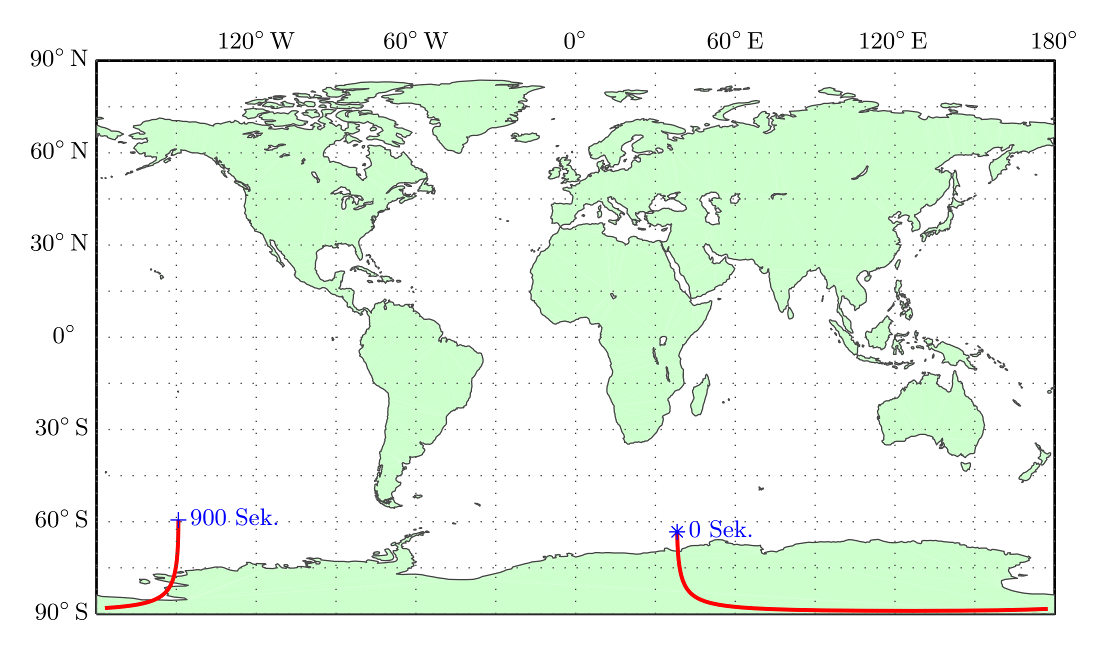


Abb. 5.8: `fig08` – Bodenspur des GRACE-A-Satelliten (Datensatz vom 01.01.2003)

5.3.2 Verarbeitung 2: kurzer Bahnabschnitt mit leicht angepassten Prozessierungsparametern

Für die zweite Verarbeitung wurden im GUI des Hauptprogramms die entsprechenden in Tabelle 5.3 aufgeführten Optionen ausgewählt. Die resultierenden Differenzen zwischen den verdichteten Orbits und der Navigationslösung (s. Abb. 5.9 und 5.10) sind nun weitaus geringer im Vergleich zur ersten Verarbeitung.

Außerdem fallen jetzt die Standardabweichungen der verdichteten Bahnvektoren am Anfang nicht mehr derart ab, siehe Abb. 5.11 und 5.11. Zweifellos erinnern diese Grafiken stark an ‚Sägezahn'-Verläufe, auch wenn die einzelnen Flanken hier keine Geraden sind. Dies ist ein durchaus typisches Ergebnis bei der Kalman-Filterung und lässt sich auch sofort begreifen, wenn man bedenkt, dass bei jedem Korrekturschritt gemäß Vorschrift (3.18) die Kovarianzen minimiert werden. Wegen der größeren Werte für `qpos` und `qvel` steigen die Werte in den 60-Sekunden-Intervallen stärker an als zuvor.

Beim K-Band-Vergleich zeigt sich, dass sich die Differenzen $\Delta\dot{\rho}$ zwischen den K-Band-Messungen und den aus den verdichteten Orbits berechneten Relativbeschleunigungen deutlich verringert haben im Vergleich zur ersten Verarbeitung (s. Abb. 5.13 unten). Allerdings treten hier bedingt durch die Messungsupdates alle 60 Sekunden teilweise größere Abweichungen bzw. Spitzen auf, die von der übrigen Größenordnung 10^{-10} m/s hervorstechen und bis in die Größenordnung 10^{-8} m/s reichen. Sie lassen sich in ihrer Amplitude nochmals verringern, indem man die relevanten Parameter erneut entsprechend abändert (s. nächste Verarbeitung).

Tab. 5.5: K-Band-Vergleich zur zweiten Verarbeitung

(a) Navigationslösungen vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.002597682	0.005025348	0.002341338
$\Delta\dot{\rho}$	(m/s)	$-1.70208 \cdot 10^{-5}$	$2.80200 \cdot 10^{-5}$	$1.75543 \cdot 10^{-5}$

(b) Berechnete Orbits vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.003508509	0.007823808	0.003126733
$\Delta\dot{\rho}$	(m/s)	$-7.97859 \cdot 10^{-9}$	$1.45493 \cdot 10^{-8}$	$2.77496 \cdot 10^{-9}$

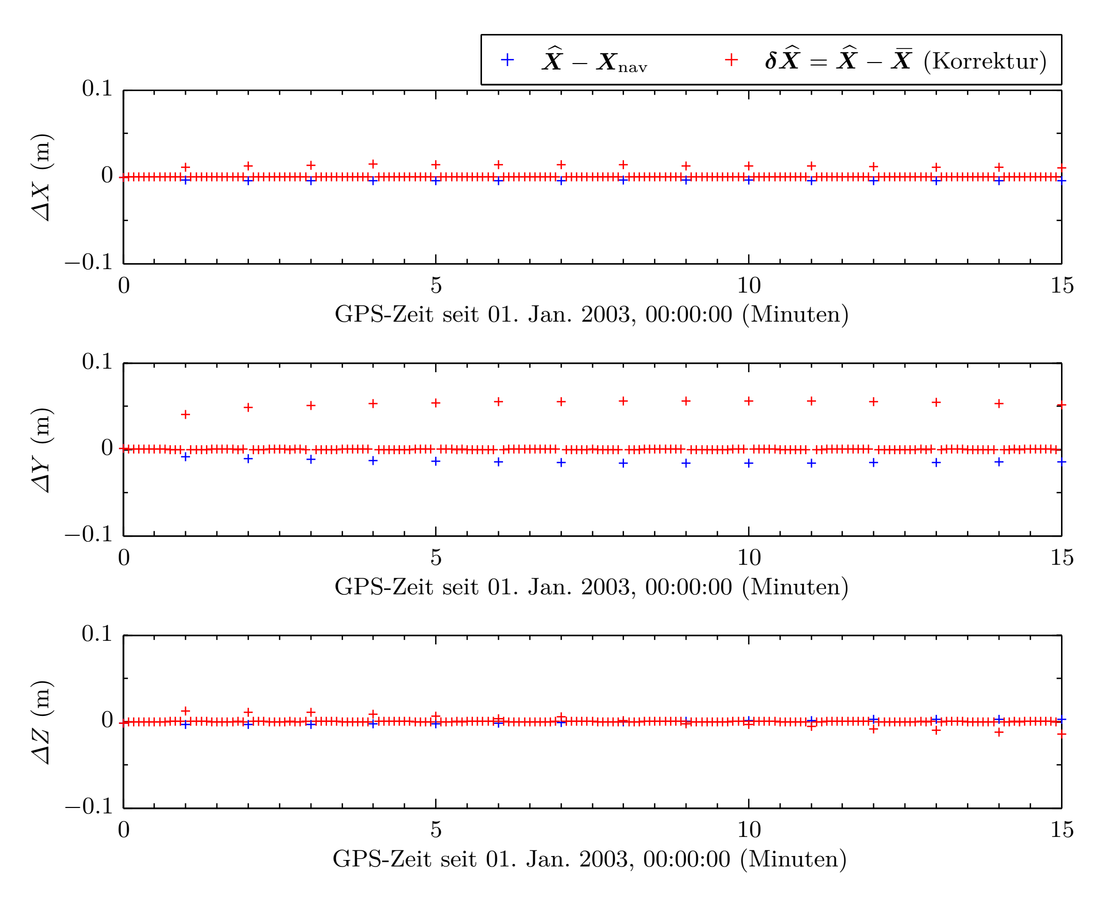


Abb. 5.9: `fig01` – geschätzte Korrekturen und Vergleich zur Navigationslösung (Positionen, GRACE A)

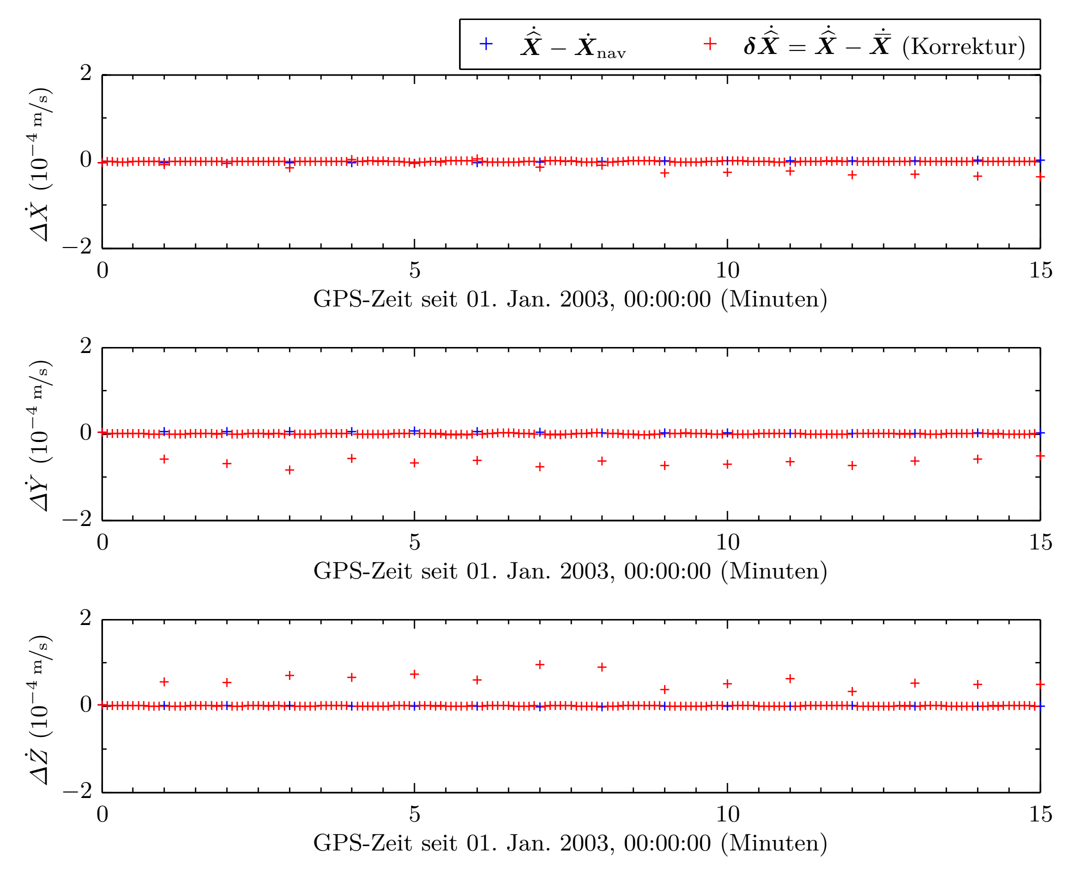


Abb. 5.10: `fig02` – geschätzte Korrekturen u. Vergl. zur Navigationslösung (Geschwindigk., GRACE A)

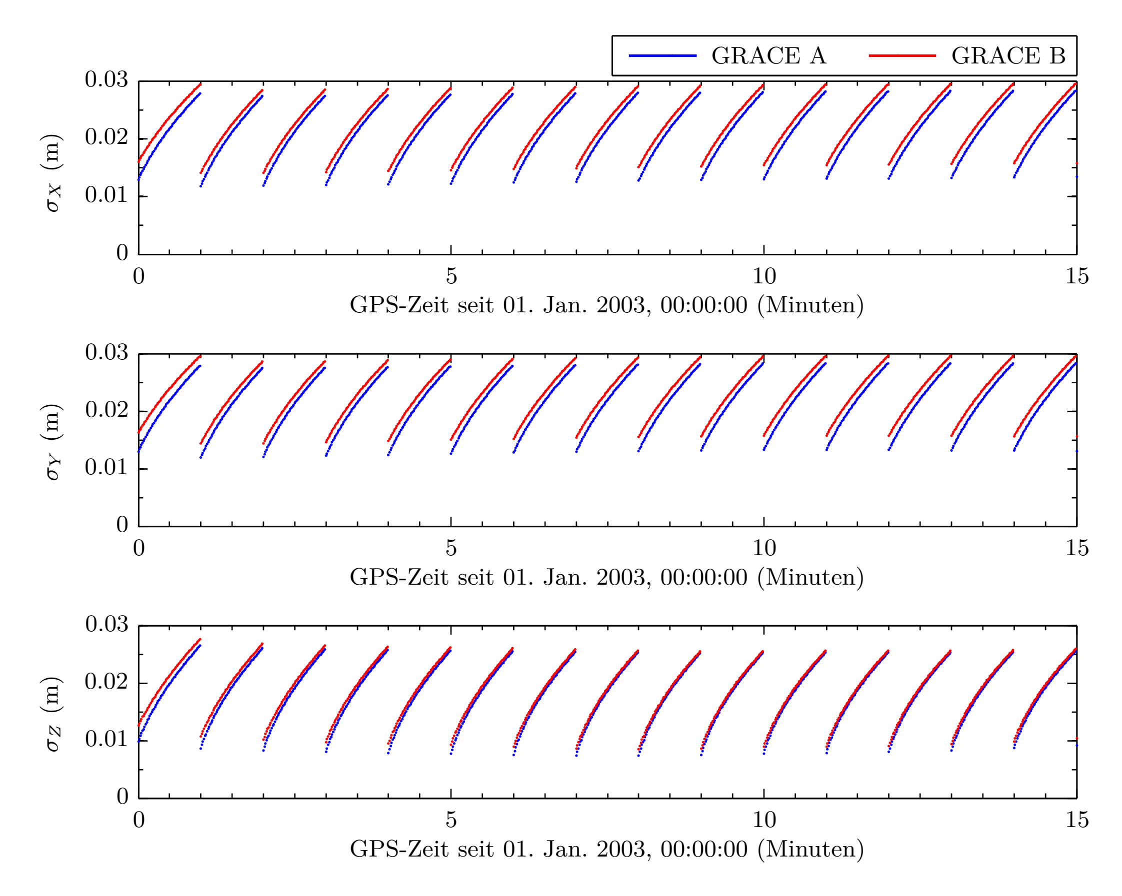


Abb. 5.11: `fig05` – Standardabweichungen der Positionskomponenten (GRACE A & B)

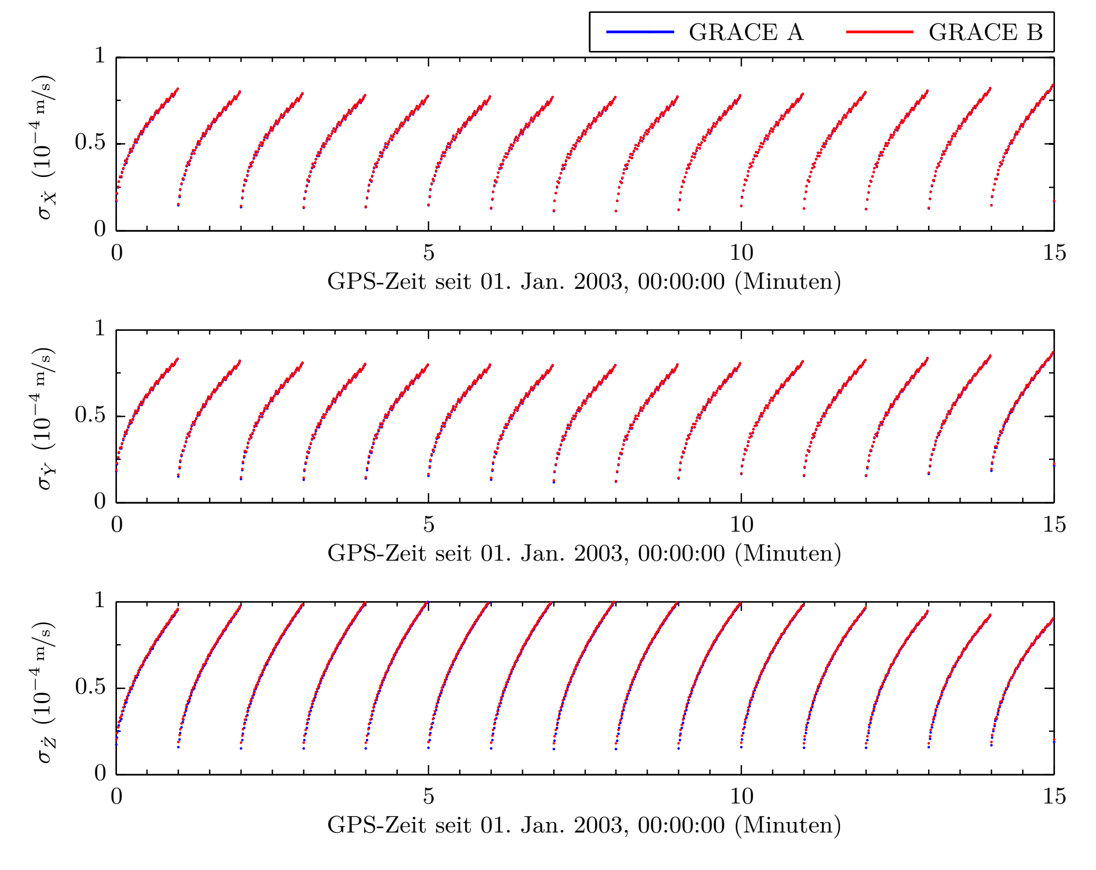


Abb. 5.12: `fig06` – Standardabweichungen der Geschwindigkeitskomponenten (GRACE A & B)

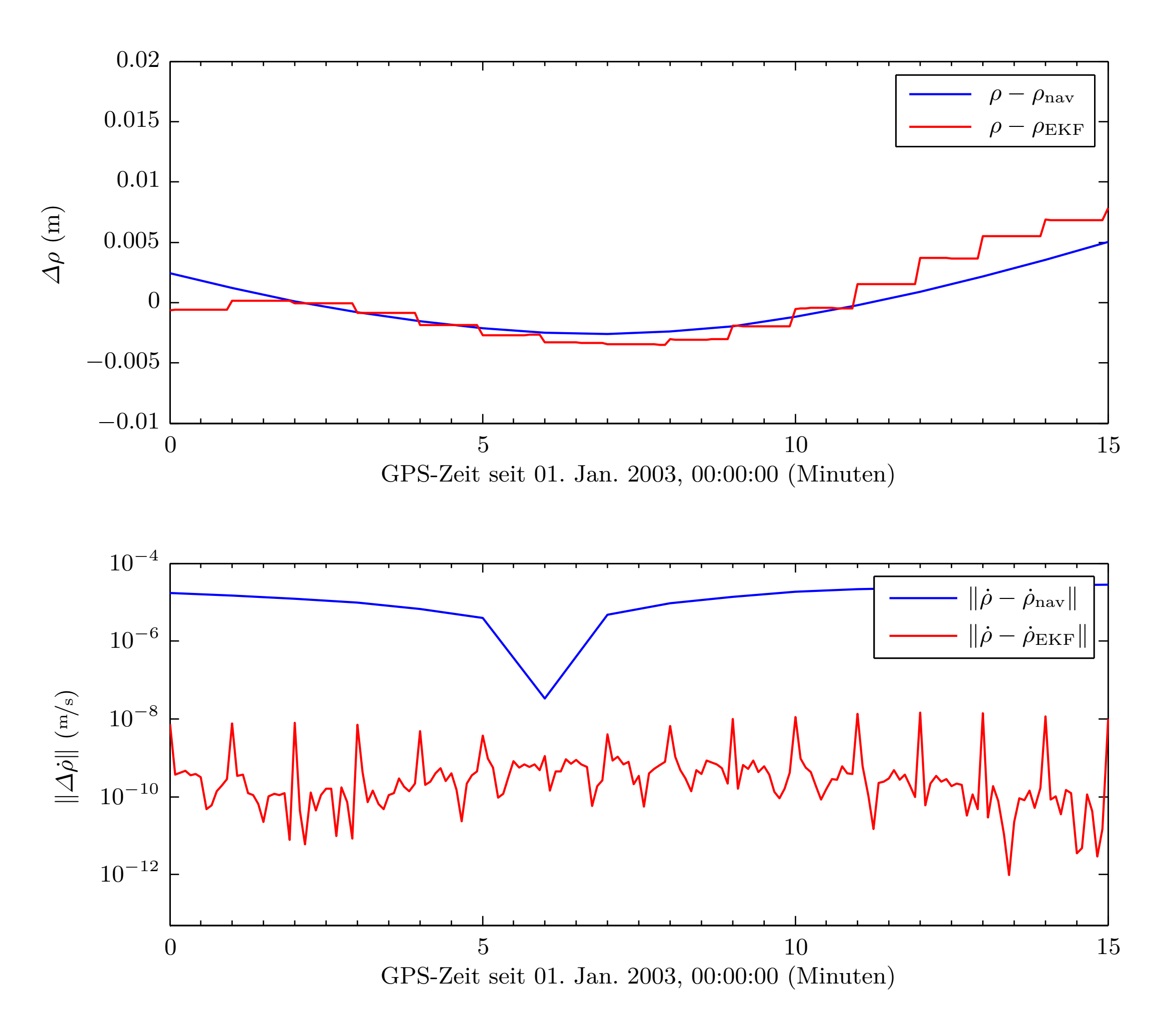


Abb. 5.13: `fig07b` – K-Band-Vergleich

5.3.3 Verarbeitung 3: kurzer Bahnabschnitt mit stärkerer K-Band-Gewichtung

Nach der dritten Prozessierung sind in den MATLAB-Grafiken `fig01` bis `fig06` keine wesentlichen Unterschiede im Vergleich zur letzten Verarbeitung erkennbar, weshalb diese hier nicht dargestellt sind. Allerdings fällt nun wie erwartet der K-Band-Vergleich in der Grafik `fig07b` besser aus, s. Abb. 5.14, da sich bei den Differenzen $\Delta\dot{\rho}$ die Spitzen reduziert haben und nun um $10^{-10}\,\mathrm{m/s}$ liegen. Die veränderten Werte für die Parameter `qvel`, $\sigma_{\dot{\rho}}$ und $\sigma_{0,\mathtt{navvel}}$ haben letztlich die Gewichtung bei den Messungsupdates noch stärker hin zu den K-Band-Beobachtungen $\dot{\rho}$ verlagert und bewirken hier somit insgesamt kleinere Differenzen $\Delta\dot{\rho}$. Dies zeigt sich auch bei der berechneten Standardabweichung, siehe Tabelle 5.6b.

Als Nächstes folgt die Prozessierung mit den identischen, hier gewählten Einstellungen für den vollständigen Datensatz, d.h. bis zum Endpunkt der Orbits bei 86340 Sekunden.

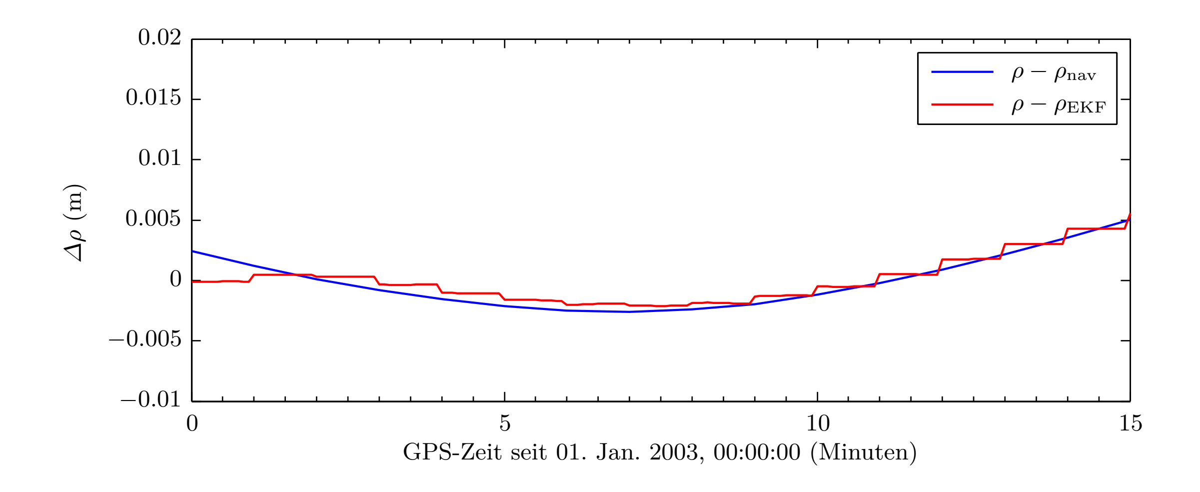
Tab. 5.6: K-Band-Vergleich zur dritten Verarbeitung

(a) Navigationslösungen vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.002597682	0.005025348	0.002341338
$\Delta\dot{\rho}$	(m/s)	$-1.70208 \cdot 10^{-5}$	$2.80200 \cdot 10^{-5}$	$1.75543 \cdot 10^{-5}$

(b) Berechnete Orbits vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.002115741	0.005564773	0.001847677
$\Delta\dot{\rho}$	(m/s)	$-2.63315 \cdot 10^{-10}$	$4.88030 \cdot 10^{-10}$	$9.04784 \cdot 10^{-11}$



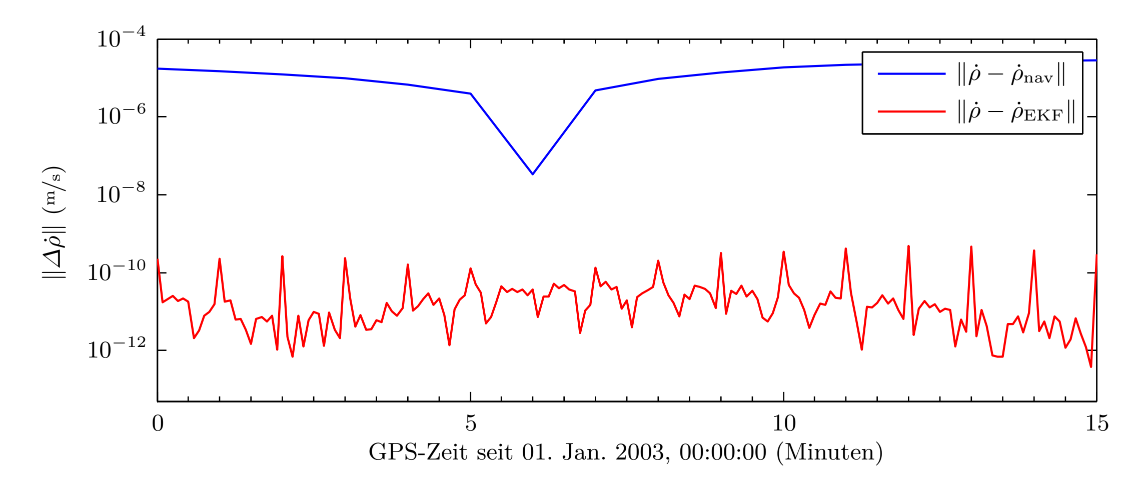


Abb. 5.14: `fig07b` – K-Band-Vergleich

5.3.4 Verarbeitung 4: Tagesorbits mit zuvor gewählten Prozessierungsparametern

Wie aus Abbildung 5.15 und 5.16 zu entnehmen ist, erreichen die Korrekturen an den 60-Sekunden-Messungsupdates bei den Positionskomponenten Werte bis etwa 6.5 cm und bei den Geschwindigkeitskomponenten Werte bis maximal 0.2 mm/s. Demgegenüber sind die an den K-Band-Messungsupdates geschätzten Korrekturen sehr klein. Sie betragen hier bei den Positionskomponenten maximal 0.703 mm (im Mittel 6.97 µm) und bei den Geschwindigkeiten maximal 0.033 mm/s (Mittelwert der Absolutbeträge 0.47 µm/s). Im Falle des zweiten GRACE-Satelliten sind die geschätzen Zuschläge ähnlich groß (s. Abb. C.5 und C.6 in Anhang C, Seite 153).

Bei dieser Orbitverdichtung über einen gesamten Tag werden erstmals mehrere Umläufe der Satelliten betrachtet, was anhand der Grafiken als sog. „1-per-revolution"-Verhalten ersichtlich ist. Denn durch Abzählen stellt man fest, dass die roten Kurven der 60-Sekunden-Korrekturen in Abbildung 5.15 circa 16 Perioden aufweisen. Anhand der Korrekturen zeigt sich also das Umlaufverhalten der Satelliten, denn pro Tag vollbringen die GRACE-Satelliten gerade etwas weniger als 16 Umläufe. Dieser kurzperiodischen Schwingung ist offenbar eine Schwingung mit einer größeren Amplitude und einer Periodendauer von etwa 12 Stunden überlagert. Man nimmt an, dass Rotationseffekte hierfür verantwortlich sind und damit auch die Größe der Korrekturen beeinflussen, denn scheinbar würde sich der Effekt bei einer Betragsbildung der drei Komponenten auslöschen (s. rote Kurven). Die Bereiche um die zwei Minima der roten Kurven lassen sich auch geographisch zuordnen, wie vor allem anhand der auf die quadratische Plattkarte projizierten Beträge der Positionskorrekturen (s. Abb. 5.21) deutlich wird. Demnach sind die betragsmäßig kleinsten Positionskorrekturen in zwei diametral gegenüberliegenden Gebieten um den Äquator bei 120° West und bei 60° Ost lokalisiert. Indes scheinen die in Abb. 5.22 dargestellten Geschwindigkeitskorrekturen weniger ortsabhängig zu sein. In der realisierten Toolbox konnten keine Fehler gefunden werden, welche die Rotationseffekte erklären würden. Eventuell liegt die Ursache also bei den verwendeten Eingangsdaten.

Im Zusammenhang mit den vorgestellten Ergebnissen muss auf einige Punkte aufmerksam gemacht werden:

- Da sich unmittelbar aus den übrigen Plattkarten-Darstellungen der Ergebnisse (MATLAB-Grafiken `fig13` bis `fig20`) keine *neuen* Schlussfolgerungen ziehen lassen, sind diese erst in Anhang C ab Seite 151 angefügt.
- Nach der vierten Prozessierung sind nun erstmals Lücken in den Ergebnissen vorhanden, wie man z.B. gut anhand der in Abb. 5.18 eingezeichneten Bodenspur erkennt. Diese Stellen wurden bei der Orbitverdichtung übersprungen, da zu den betreffenden Zeitpunkten keine K-Band-Messungen vorliegen (`NaN`-Einträge).
- Bei Minute 678 gibt es mit ca. 0.24 mm/s (bzw. 0.25 mm/s) einen vergleichsweise großen Zuschlag zur $\dot{Y}$-Komponente des prädizierten GRACE-A- (bzw. GRACE-B-) Zustandsvektors. In Abb. 5.16 und Abb. C.6 in Anhang C wurde daher der Wertebereich der Ordinatenachse eingeschränkt (s. Funktion `ekf_plot1`). Bei den ‚Scatter-Plots' auf der Plattkarte in Abb. 5.22 und Abb. C.10 (MATLAB-Grafiken `fig12` und `fig16`) beträgt dieser ‚Ausreißer' ungefähr 0.85 mm/s (bzw. 0.83 mm/s) und würde somit verhindern, dass die übrigen Werte vernünftig dargestellt werden (lineare Skalierung). Deshalb wurden die betreffenden Stellen in Abb. 5.22 und Abb. C.10 ausgelassen durch vorige Anwendung der Funktion `elim_spikes`, welche die $j \in \mathbb{N}$ größten und kleinsten Einträge eines Vektors zu `NaN` setzt (hier gewählt: $j \equiv 1$).
- Die Ausgabegrafiken `fig05` und `fig06` werden für diese Prozessierung weder hier noch in Anhang C abgebildet, da die prinzipielle Entwicklung der Schätzfehlervarianzen der Systemzustände bereits anhand Abb. 5.11 und 5.12 der vorherigen Verarbeitung klar geworden ist und die Kurven über den gesamten Tag ähnlich verlaufen. Die Werte liegen hier zwischen ungefähr 1 cm und 3.7 cm bei den Positions- und zwischen 0.04 mm/s und 0.27 mm/s bei den Geschwindigkeitskomponenten.

Besonders hervorzuheben ist das in Abbildung 5.17 gezeigte Ergebnis. Darin weisen die Relativgeschwindigkeiten, welche aus den verdichteten Bahnvektoren berechnet wurden, über die gesamte Tageslänge nur sehr geringe Differenzen zu den K-Band-Messungen auf. Auch die Standardabweichung zwischen den 5-sekündlichen K-Band-Messungen $\dot{\rho}$ und den berechneten Werten beträgt nur etwa 0.078 nm/s (vgl. Tab. 5.7b). Damit ist das wichtigste eingangs gesetzte Ziel dieser Arbeit erreicht.

Tab. 5.7: K-Band-Vergleich zur vierten Verarbeitung

(a) Navigationslösungen vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.033909544	0.043333370	0.013361674
$\Delta\dot{\rho}$	(m/s)	$-7.55870 \cdot 10^{-5}$	$6.50815 \cdot 10^{-5}$	$1.89306 \cdot 10^{-5}$

(b) Berechnete Orbits vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.033671603	0.050048328	0.014759047
$\Delta\dot{\rho}$	(m/s)	$-1.48526 \cdot 10^{-9}$	$8.48147 \cdot 10^{-10}$	$7.75754 \cdot 10^{-11}$

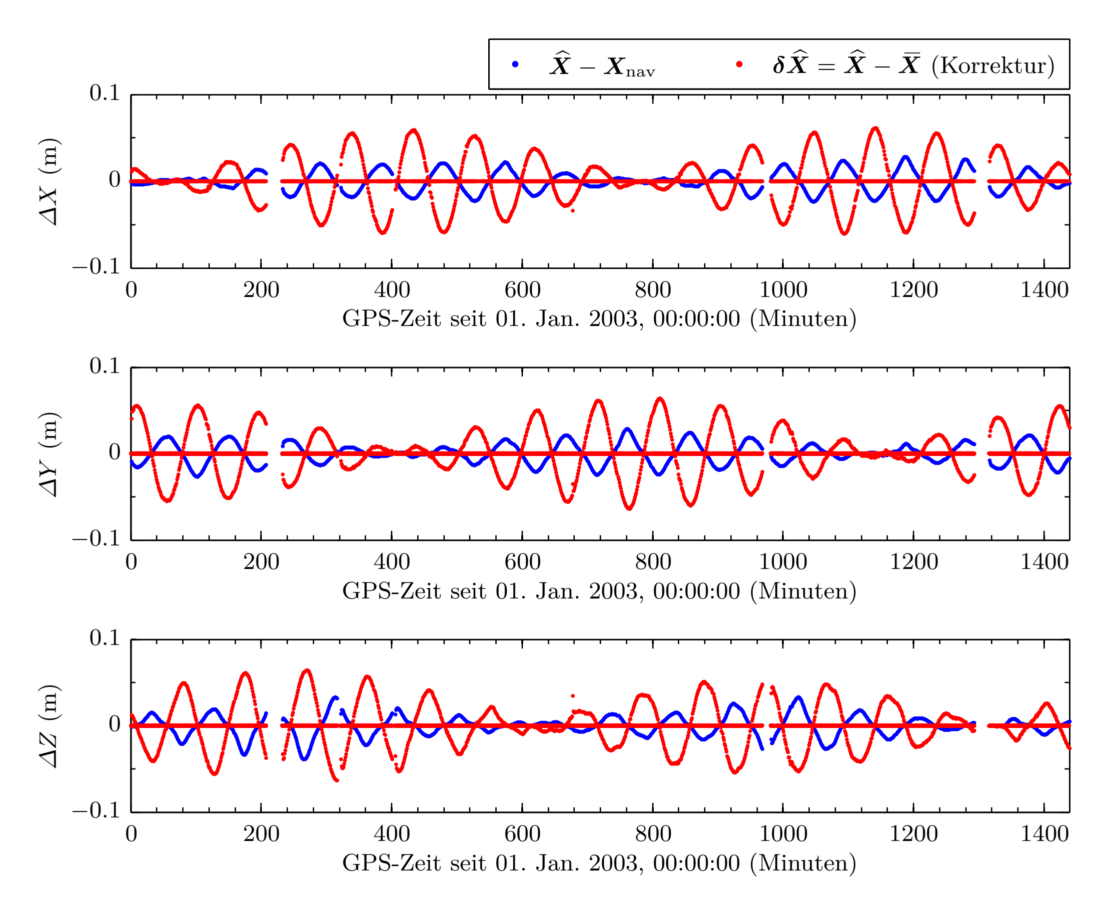


Abb. 5.15: `fig01` – geschätzte Korrekturen und Vergleich zur Navigationslösung (Positionen, GRACE A)

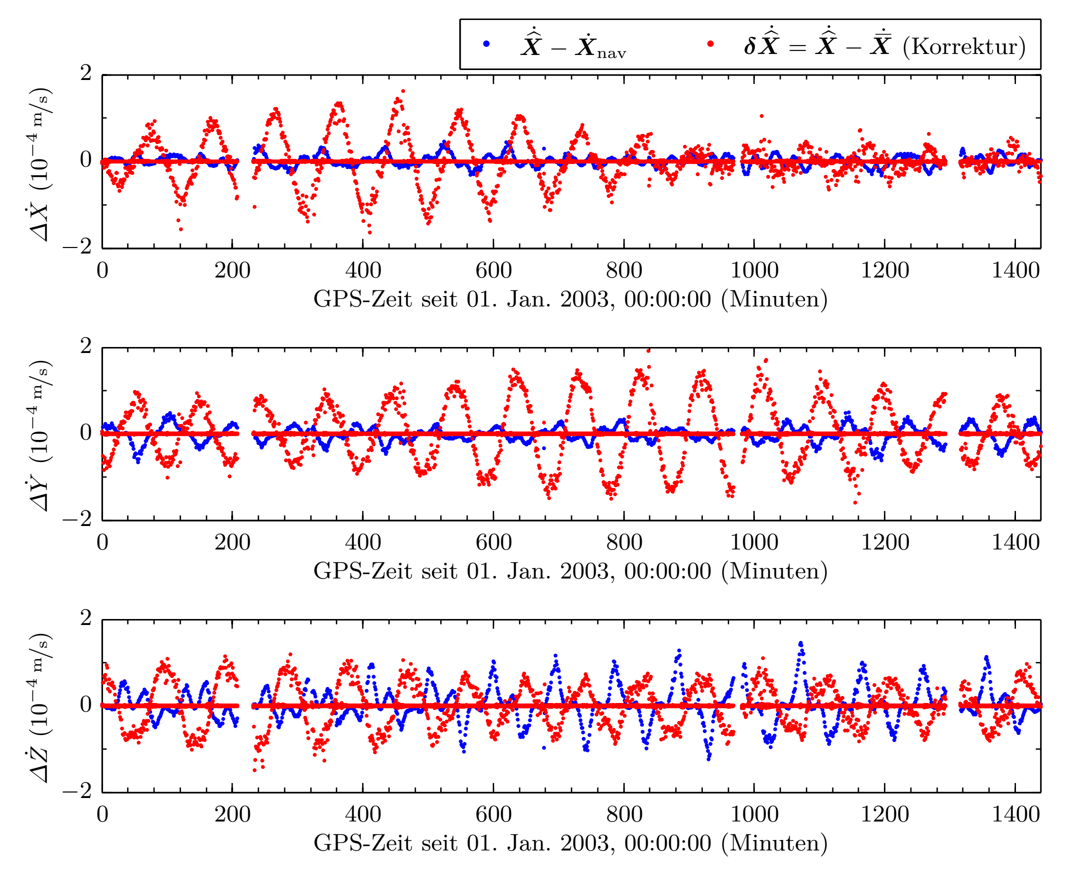


Abb. 5.16: `fig02` – geschätzte Korrekturen u. Vergl. zur Navigationslösung (Geschwindigk., GRACE A)

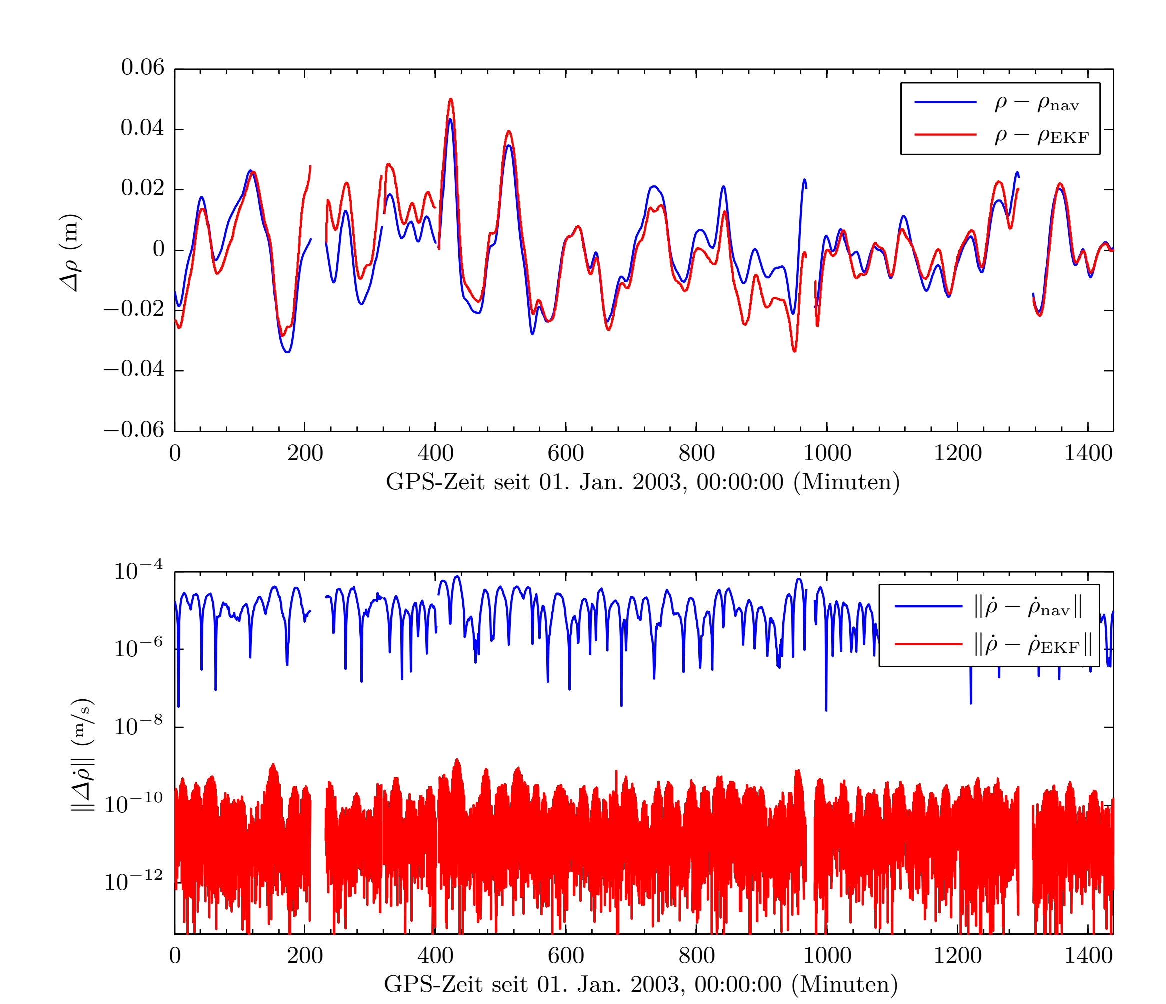


Abb. 5.17: `fig07b` – K-Band-Vergleich

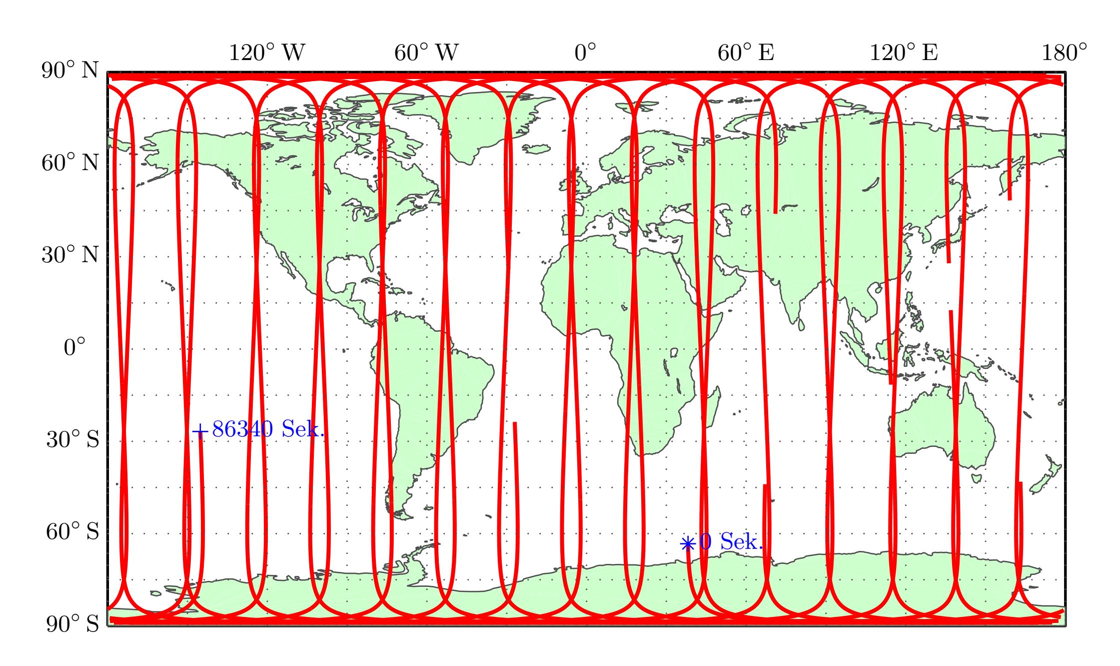


Abb. 5.18: `fig08` – Bodenspur des GRACE-A-Satelliten (Datensatz vom 01.01.2003)

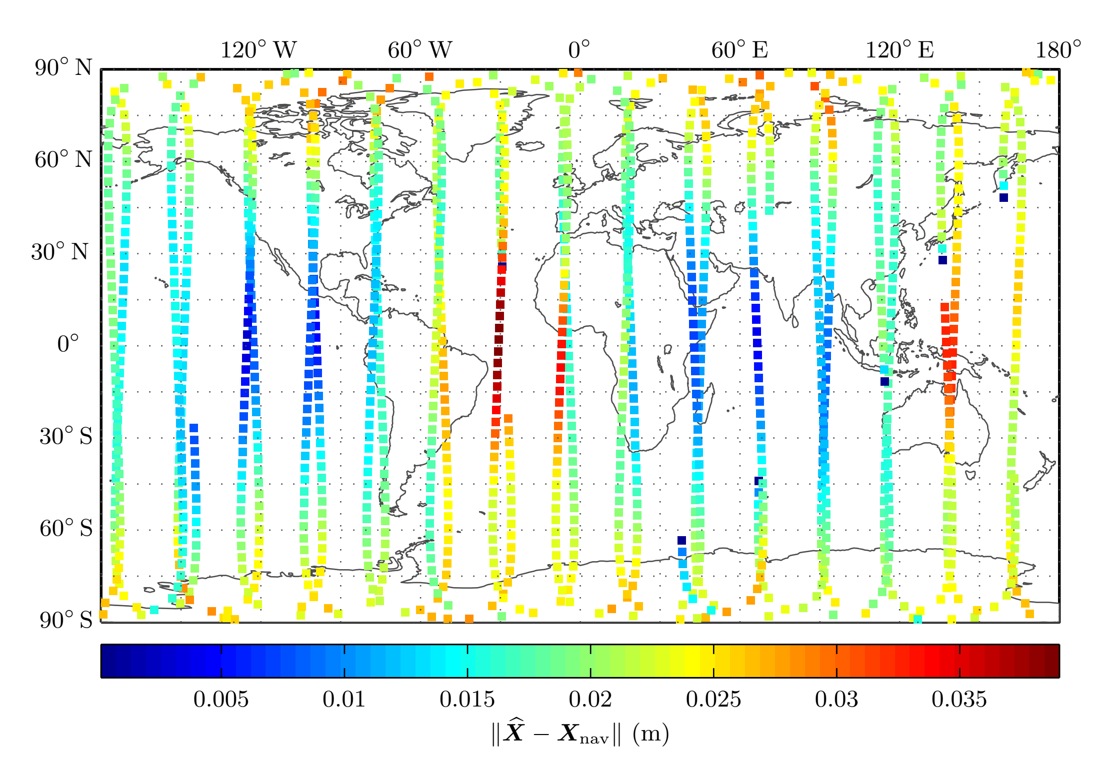


Abb. 5.19: `fig09` – berechnete Positionen vs. Navigationslösung (GRACE A)

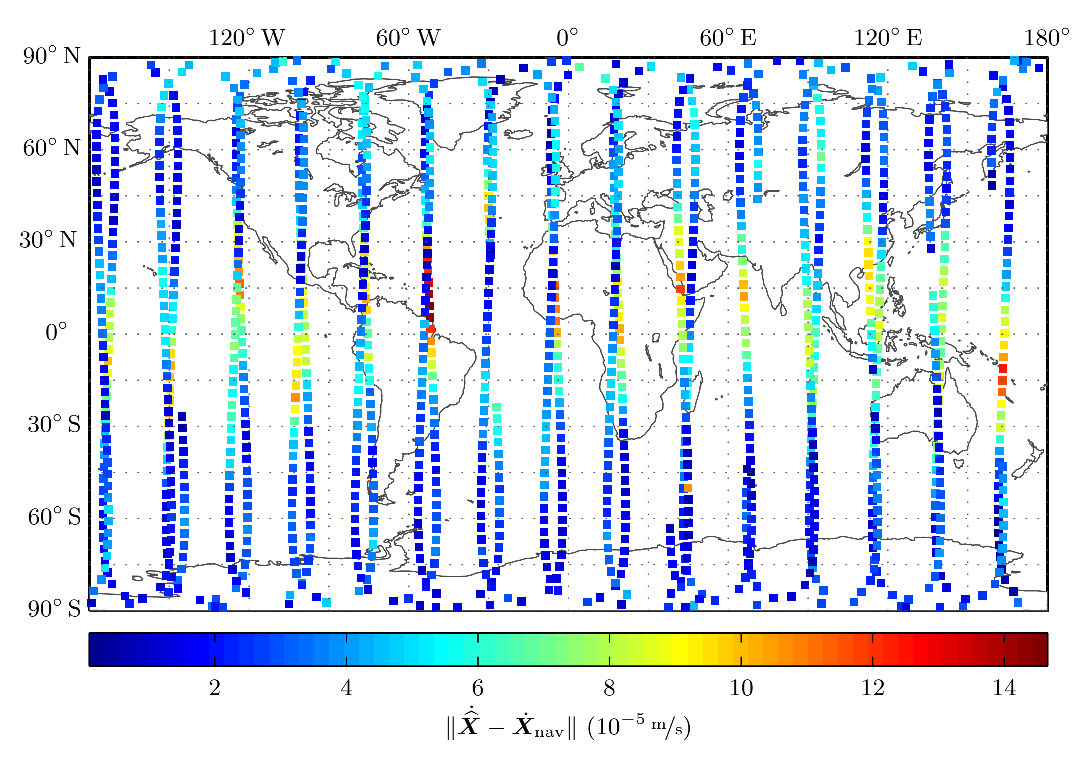


Abb. 5.20: `fig10` – berechnete Geschwindigkeiten vs. Navigationslösung (GRACE A)

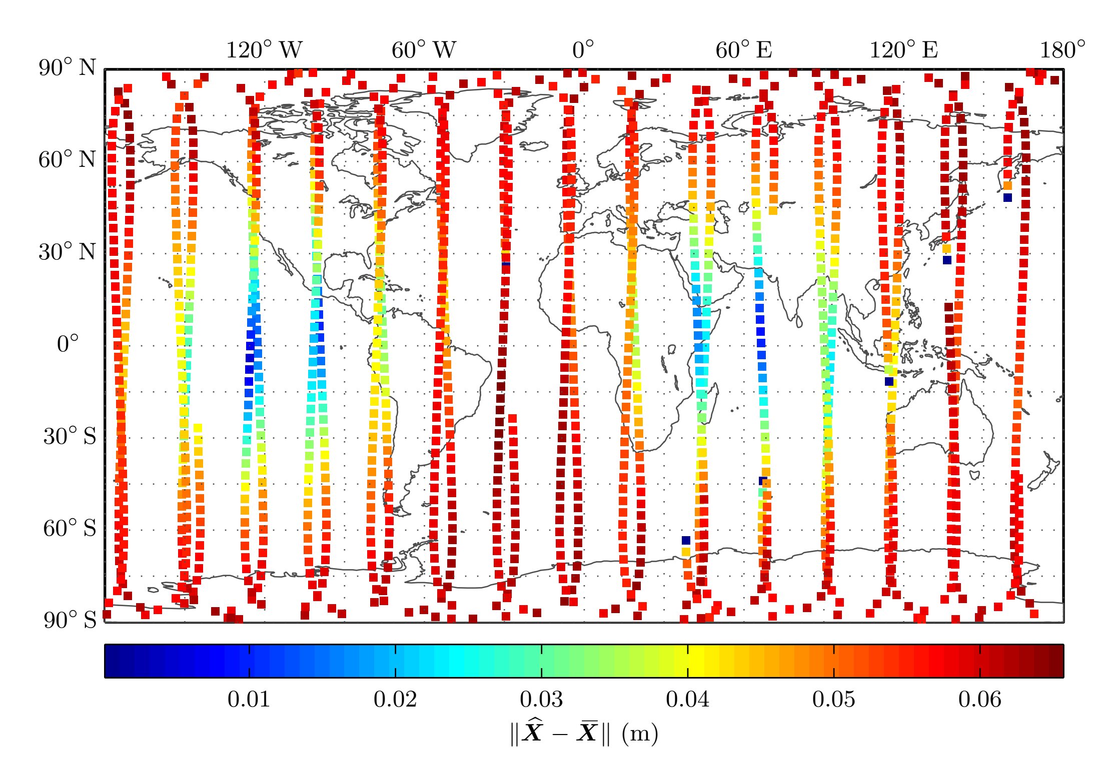


Abb. 5.21: `fig11` – geschätzte Verbesserungen der prädizierten Positionen (GRACE A)

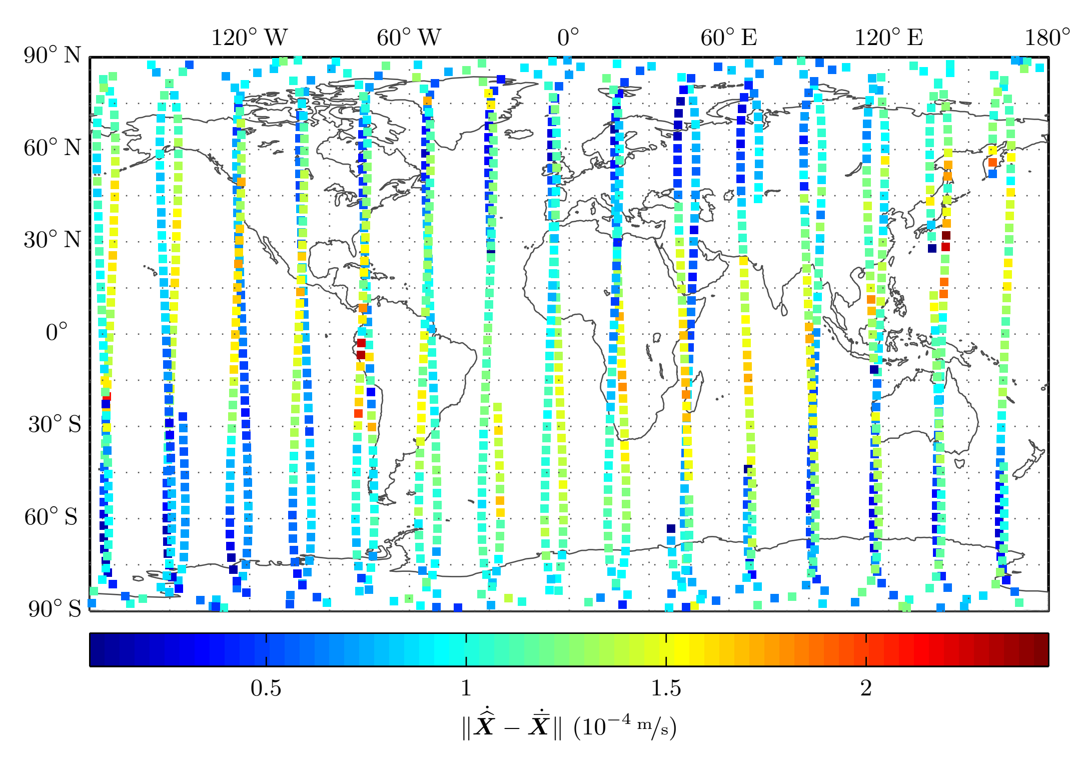


Abb. 5.22: `fig12` – geschätzte Verbesserungen der prädizierten Geschwindigkeiten (GRACE A)

5.3.5 Verarbeitung 5: Tagesorbits mit dem J_2-Integrator

Bei der fünften Orbitverdichtung der realen GRACE-Daten mittels `ekf_orbit2` wurde der einfache Integrator `orbint_j2` verwendet. Dabei ist die Frage von Interesse, inwiefern die resultierenden Orbits beim anschließenden K-Band-Vergleich noch in Einklang stehen zu den gemessenen Relativgeschwindigkeiten $\dot{\rho}$.

Aufgrund des einfachen Kräftemodells wurden größere Parameterwerte für `qpos` und `qvel` gewählt als bei den vorigen Verdichtungen. Erfreulicherweise liegen nun Differenzen $\Delta\dot{\rho}$ vor, die tatsächlich zumeist kleiner als 10^{-8} m/s sind, wie die Ergebnisse in Abbildung 5.25 zeigen. Damit wird die bereits erwähnte Fähigkeit des umgesetzten Kalman-Filters bestätigt, wonach sich trotz einfacher Kräftemodelle eine gute *relative* Orbitgenauigkeit, d.h. eine hohe Genauigkeit in Bezug auf die Relativgeschwindigkeiten $\dot{\rho}$, erzielen lässt.

Doch die übrigen Resultate bei dieser Variante der Orbitverdichtung mit dem J_2-Integrator sind, wie zu erwarten war, wenig zufriedenstellend. Einzig die kurze Rechenzeit von nur etwa fünfeinhalb Minuten für den gesamten Datensatz ist als Vorteil zu nennen. Weitaus stärker wiegt jedoch das Manko, dass die prädizierten Zustandsvektoren aufgrund des einfachen Kräftemodells von der entsprechenden Navigationslösung beträchtlich abweichen. So betragen die Positionskorrekturen nun beinahe schon einen Meter, siehe Abbildung 5.23.

Insgesamt ist wegen der vergleichsweise großen Korrekturen davon abzuraten, den einfachen Orbitintegrator `orbint_j2` oder gar die Integratoren `orbint_kep(2)` bei der Verdichtung der realen GRACE-Navigationslösungen zu nutzen. Stattdessen sind hierfür die Funktionen `orbint_field(2)` zu bevorzugen, da die damit integrierten Bahnen wesentlich besser zu den gegebenen Navigationslösungen passen (kleinere Korrekturen).

Tab. 5.8: K-Band-Vergleich zur fünften Verarbeitung

(a) Navigationslösungen vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.033909544	0.043333370	0.013361674
$\Delta\dot{\rho}$	(m/s)	$-7.55870 \cdot 10^{-5}$	$6.50815 \cdot 10^{-5}$	$1.89306 \cdot 10^{-5}$

(b) Berechnete Orbits vs. K-Band

		Minimum	Maximum	Standardabweichung
$\Delta\rho$	(m)	-0.061754253	0.081157026	0.016402822
$\Delta\dot{\rho}$	(m/s)	$-1.47867 \cdot 10^{-8}$	$1.21337 \cdot 10^{-8}$	$7.82807 \cdot 10^{-10}$

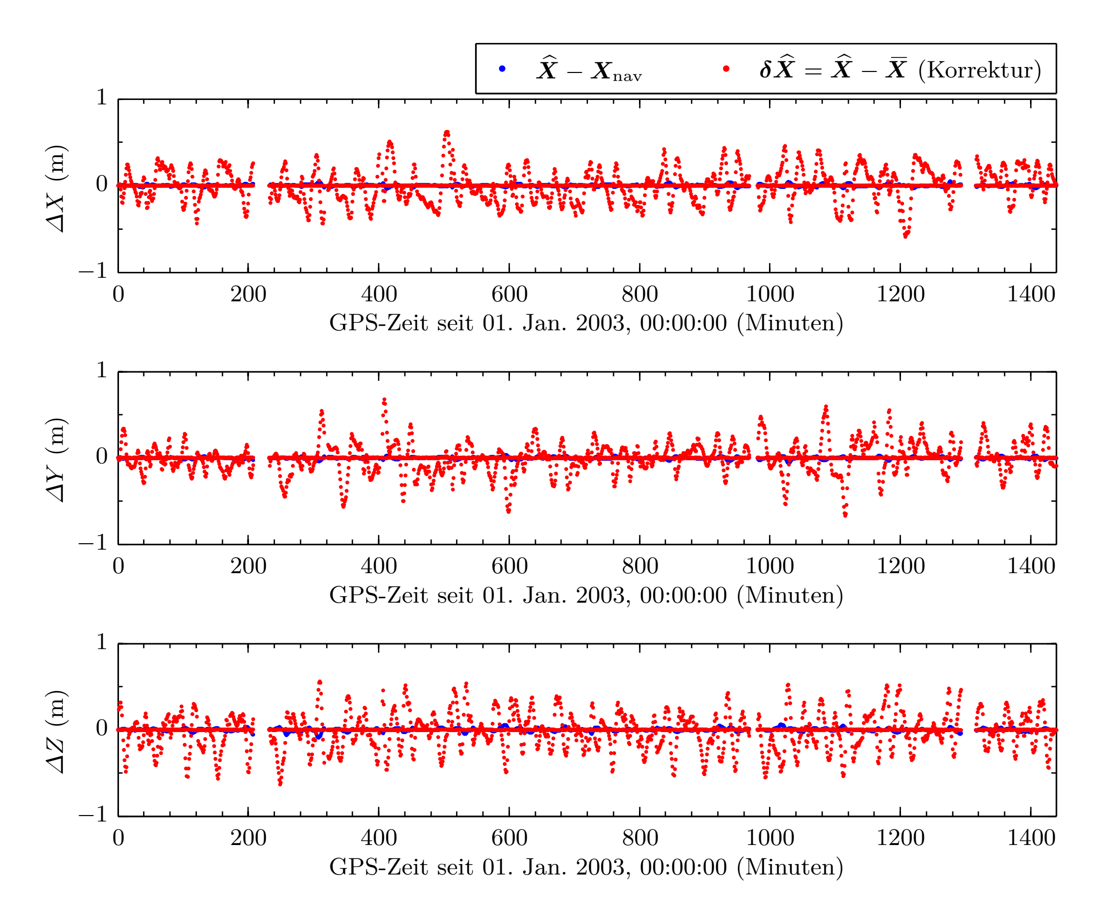


Abb. 5.23: `fig01` – geschätzte Korrekturen und Vergleich zur Navigationslösung (Positionen, GRACE A)

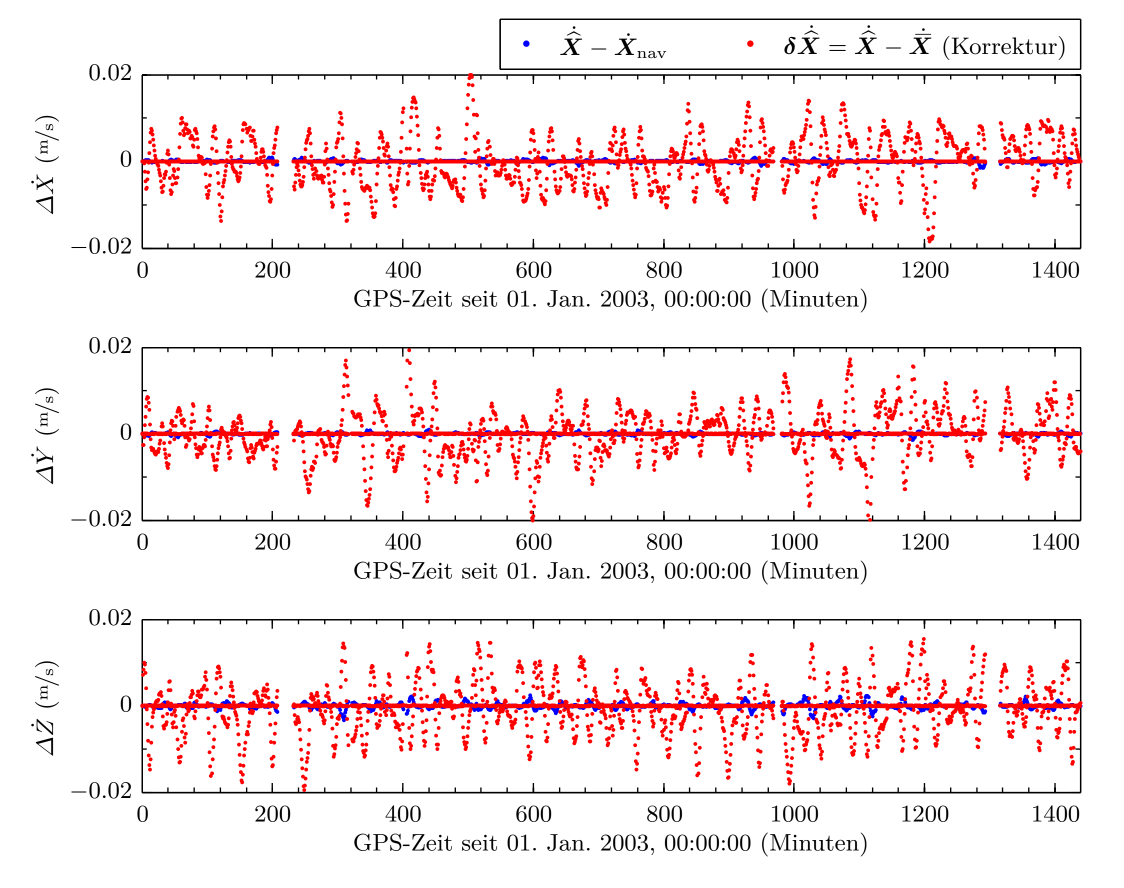


Abb. 5.24: `fig02` – geschätzte Korrekturen u. Vergl. zur Navigationslösung (Geschwindigk., GRACE A)

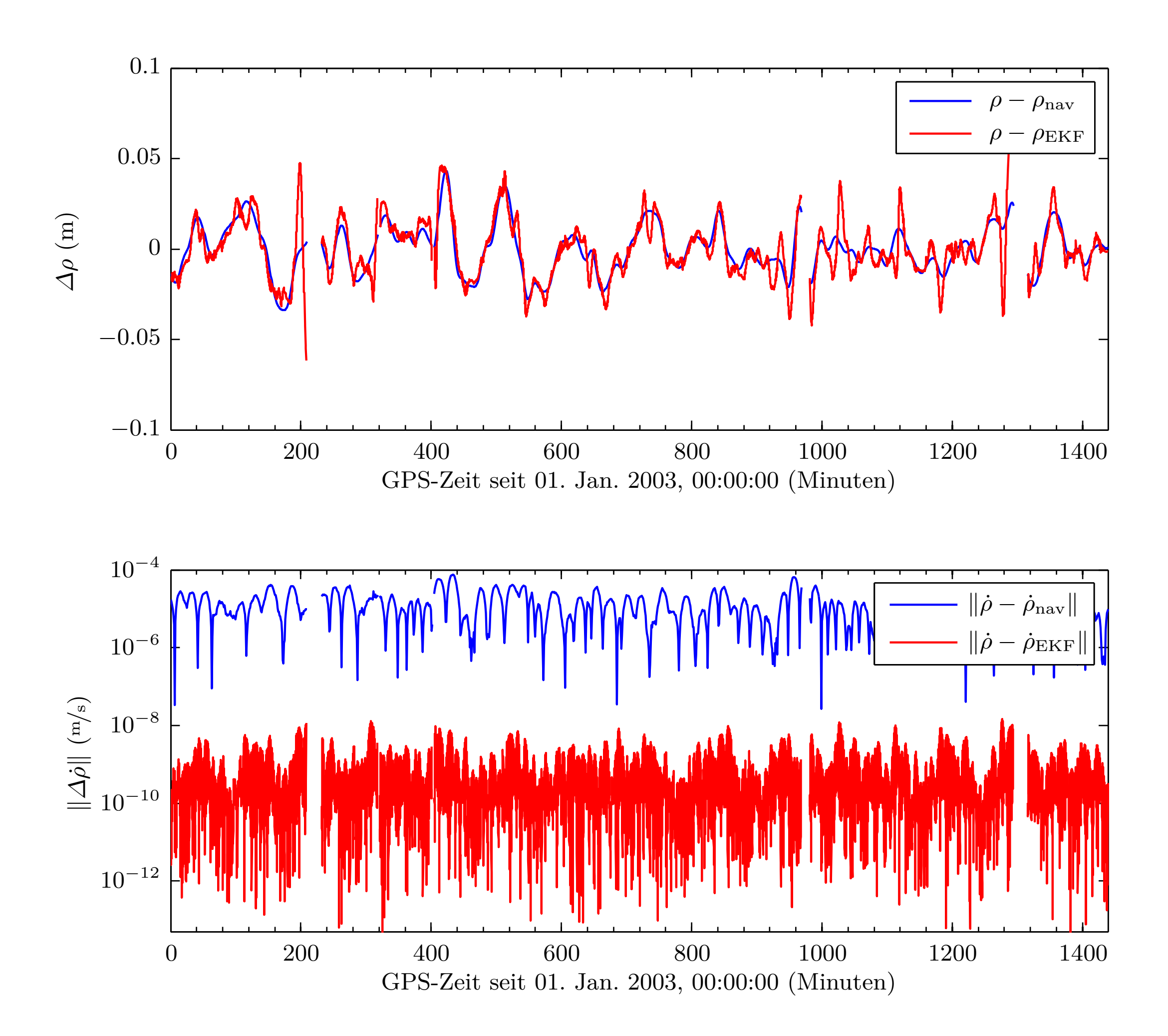


Abb. 5.25: `fig07b` – K-Band-Vergleich

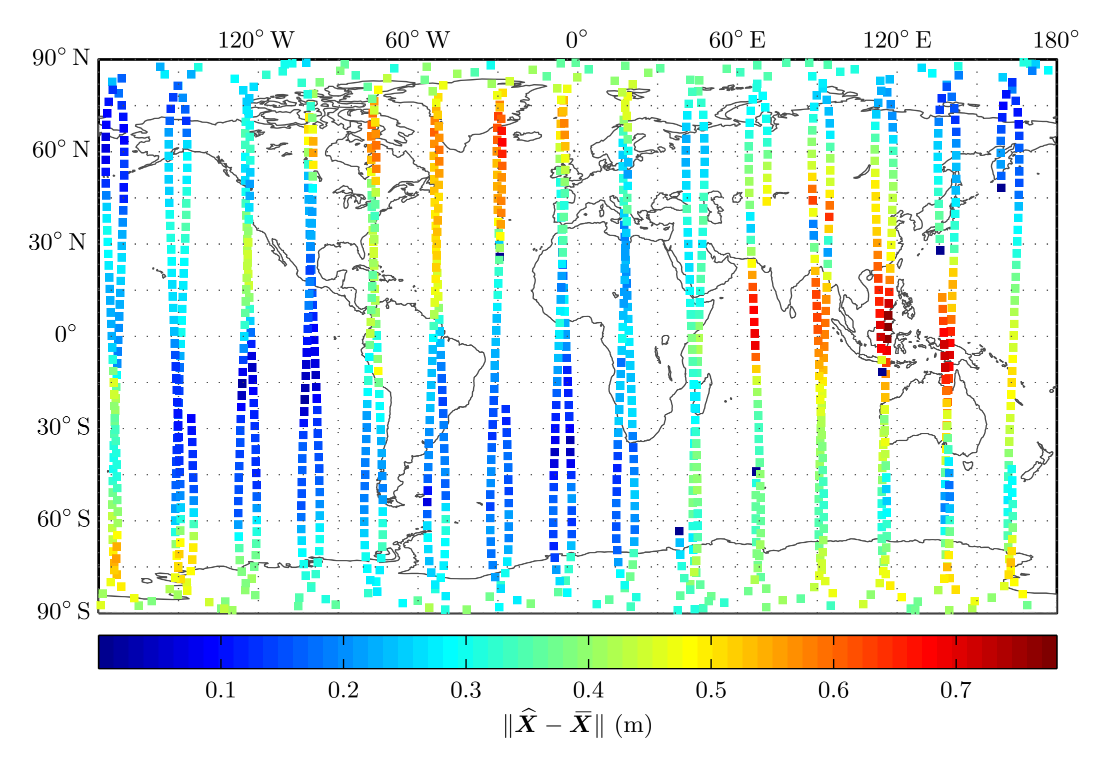


Abb. 5.26: `fig11` – geschätzte Verbesserungen der prädizierten Positionen (GRACE A)

5.3.6 Nachprozessierung der verdichteten Orbits

In Abschnitt 3.2.7 wurde bereits auf den Umstand hingewiesen, dass bei den Messungsupdates zwangsläufig ein Sprung zwischen dem zuletzt prädizierten und dem korrigierten Systemzustand entsteht, welcher gerade der geschätzten Korrektur entspricht. Dementsprechend weisen auch die bislang mit der Toolbox verdichteten GRACE-Orbits zwischen den einzelnen Bahnbogenstücken Sprünge auf, wobei, wie oben in Verarbeitung 4 (Abschn. 5.3.4) quantifiziert wurde, alle 60 Sekunden ein größerer Zuschlag bzw. Sprung und dazwischen in 5-Sekunden-Abständen kleinere Korrekturen bzw. Sprünge auftreten. Die Unstetigkeiten zeigen sich auch beim Vergleich mit den gemessenen Relativentfernungen ρ. Hier liegen die durch das 60-Sekunden-Update verursachten Sprünge meist im Bereich von einem Millimeter (vgl. z.B. Abb. 5.14).

Nachfolgend werden drei einfache Techniken zur nachträglichen Glättung der verdichteten Orbits untersucht. Bei allen diesen Methoden kommt es jedoch zu einem grundsätzlichen Konflikt, weshalb die vorgestellten Methoden bestenfalls als ‚experimentell' gelten können: möchte man gewährleisten, dass die Orbits weiterhin ebensogut mit der K-Band-Beobachtung $\dot{\rho}$ übereinstimmen, kommt man nicht umhin, alle beim K-Band-Update korrigierten Zustandsvektoren, die fünf Sekunden auseinander liegen, beizubehalten und nur zwischen diesen zu interpolieren. Da gerade bei 60 Sekunden wegen des Updates durch die Navigationslösung der größte Spung vorliegt, erhält man im 5-Sekunden-Intervall unmittelbar davor einen unverhältnismäßig steilen Anstieg, der im Sinne einer Glättung eigentlich gleichmäßig über das gesamte 60-Sekunden-Intervall verteilt werden müsste (siehe Abb. 5.27).

Zur Auflösung des Konflikts wäre eine Fehlerfortpflanzung für die jeweilige Methode zu entwickeln, damit nach erfolgter Glättung erneut eine Kalman-Filterung nur unter Verwendung der K-Band-Messungen $\dot{\rho}$ möglich ist. Diese erneute Kalman-Filterung ist nötig, weil die über längere Intervalle geglätteten Orbits (z.B. 60 Sekunden) nicht mehr hinreichend zu den Beobachtungen $\dot{\rho}$ passen. Wie man bei der vierten Verarbeitung in Abschn. 5.3.4 gesehen hat, sind die beim K-Band-Messungsupdate geschätzten Korrekturen vernachlässigbar klein, sodass die durch die erneute Kalman-Filterung entstehenden Sprünge im geglätteten Orbit ebenfalls zu vernachlässigen wären. Vermutlich ist aber die Entwicklung einer Fehlerfortpflanzung für eine der drei Glättungsmethoden mit ebensoviel Aufwand verbunden wie die Umsetzung eines Glättungsfilters, weshalb hier beides nicht mehr realisiert werden kann.

Am Beispiel der in Abschnitt 5.3.3 verdichteten 15-Minuten-Orbits (Verarbeitung 3) werden als Glättungsmethoden die kubische Spline-Interpolation, das Einfach-Schießverfahren sowie die kleinste-Quadrate-Einpassung in insgesamt fünf Tests untersucht (s. Tab. 5.9).

Tab. 5.9: Übersicht der durchgeführten Nachprozessierungen

Test Nr.	Methode	Inkrement n	Zeitintervall (Sek.)	`TolX`	`TolFun`
1	Kub. Splines	5	–	–	–
2	Kub. Splines	10	–	–	–
3	`fsolve`	5	5	10^{-6}	10^{-12}
4	`lsqnonlin`	5	60	10^{-7}	10^{-10}
5	`lsqnonlin`	1	900	10^{-7}	10^{-10}

Methode 1 – Kubische Spline-Interpolation

Zunächst bietet sich zur Glättung die schon aus dem zweiten Kapitel bekannte Interpolation mittels kubischer Splines an. Falls man im Optionen-GUI des Hauptprogramms (Abb. 4.2, S. 65) unter Punkt ⑨ diese Methode auswählt, hat nur die Einstellung für das Inkrement n einen Einfluss auf das Ergebnis, die restlichen Optionen (u.a. `TolX`, `TolFun`) sind erst für die nachfolgenden Methoden

gedacht. Der Parameter $n \in \mathbb{N}$ stellt den Zuwachs dar in der Anweisung `1:n:end`, welche zur Auswahl der zuvor verdichteten Bahnvektoren dient. D.h. bei der Nachprozessierung wird nur jeder n-te Zustandsvektor der gefilterten Orbits verwendet. Indem man im ersten Test n zu 5 setzt, wählt man also bei dieser Methode alle diejenigen Zustandsvektoren als Stützpunkte aus, die 5 Sekunden auseinanderliegen.

Im Ergebnis ist folgerichtig der Vergleich der geglätteten Orbits mit den K-Band-Messungen völlig identisch zum vorherigen K-Band-Vergleich in Abb. 5.14. Allerdings findet die Glättung der größeren, 60-sekündlich auftretenden Sprünge nur in einem kurzen Zeitfenster von 5 bis 10 Sekunden statt, weshalb die Interpolationskurve dort stark ansteigt (s. Abb. 5.27).

Bereits mit $n = 10$ (Test 2) fällt der Vergleich mit den K-Band-Messungen $\dot{\rho}$ um vier Größenordnungen schlechter aus (s. Abb. 5.28). Das erwähnte Zeitfenster vor den 60-Sekunden-Sprüngen erweitert sich entsprechend zu 10 bis 20 Sekunden (ohne Abb.). Mit noch höheren Werten für n werden die Abweichungen zum gefilterten Orbit schlicht zu groß (bei $n = 60$ bis zu mehreren Metern, wie schon in Abb. 2.5, S. 18).

Abschließend lässt sich diese Glättungsmethode, sofern $n = 5$ gewählt wurde, als relativ ‚gefahrlos' einstufen, denn die Abweichung der geglätteten Bahnkurve von der gegebenen überschreitet an keiner Stelle die Größe des vorhandenen Sprungs.

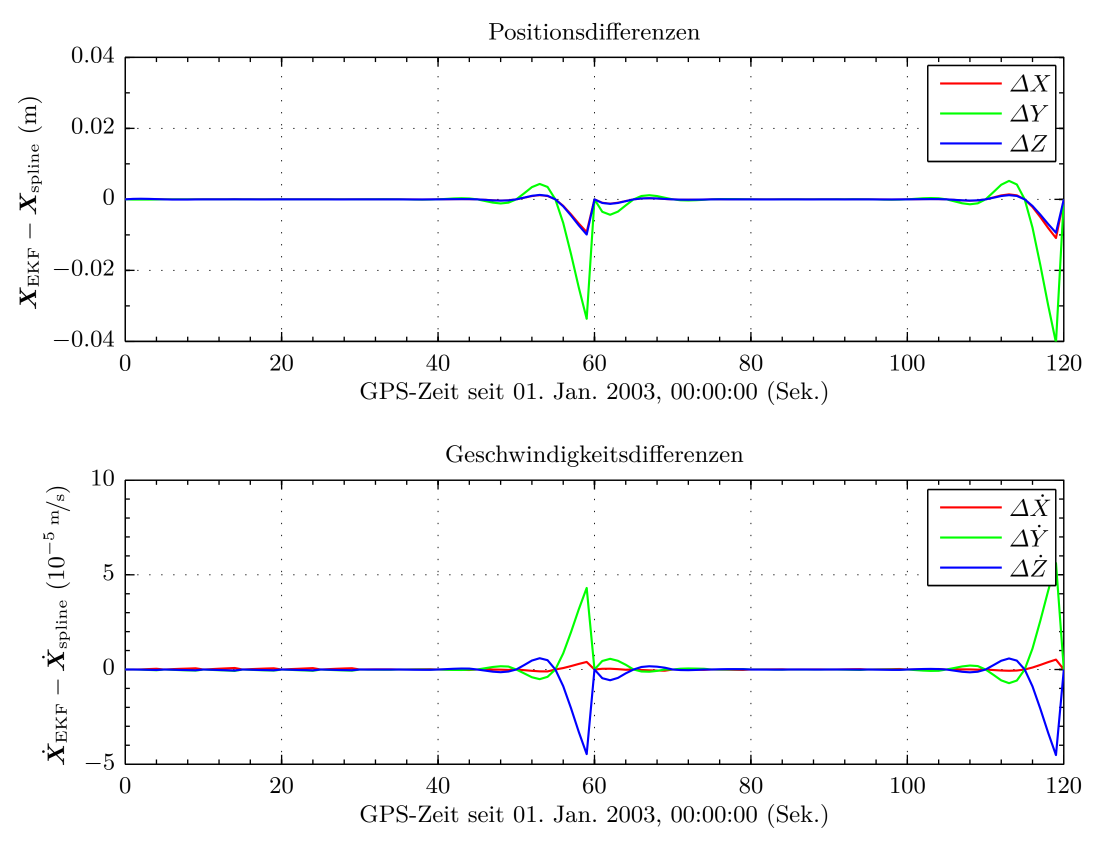


Abb. 5.27: Gefilterter vs. geglätteter GRACE-A-Orbit über 120 Sek. in Test 1

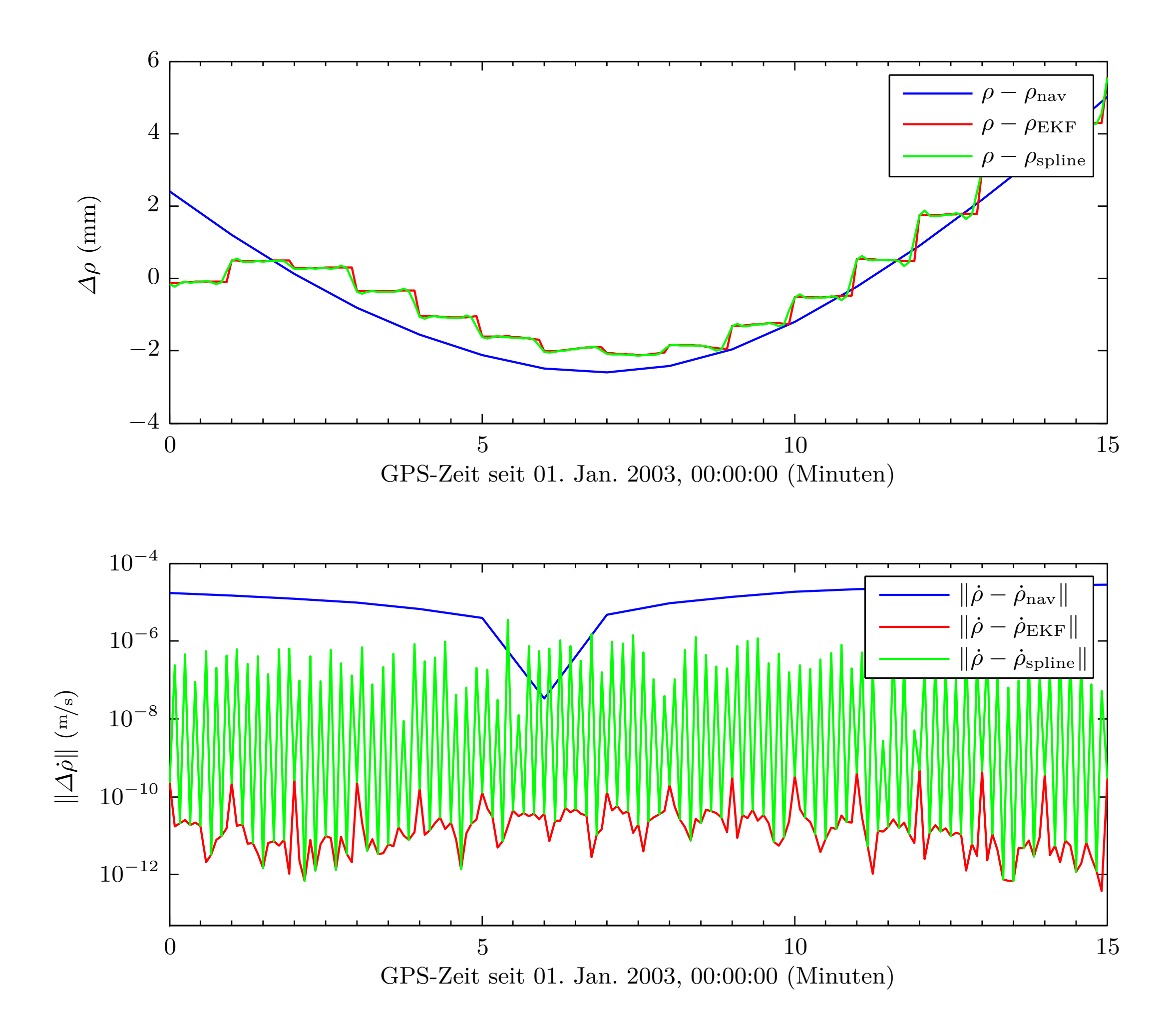


Abb. 5.28: K-Band-Vergleich zu Test 2

Methode 2 – Schießverfahren mittels `fsolve`

Die Idee ist hierbei, zwei aufeinanderfolgende, durch die Messungsupdates korrigierte Zustandsvektoren für die Formulierung eines RWPs zu nutzen. Dessen Lösung soll durch das in Abschnitt 2.3.4 beschriebene Einfach-Schießverfahren erfolgen.

Neben dem Inkrement n ist im Optionen-GUI (Abb. 4.2) dazu das Zeitintervall zwischen den Randpunkten vorzugeben. Die Lösung für *ein* Zeitintervall nach dieser und der nächsten Methode liefert die implementierte Funktion `shootorb` (s.a. Anhang A.2, S. 142).

Möchte man z.B. die Positionskomponenten des linken und rechten Randpunkts fixieren, korrigiert die Funktion mittels der `fsolve`-Routine entsprechend der im zweiten Kapitel beschriebenen Arbeitsweise die Geschwindigkeitskomponenten des linken Randpunkts iterativ, um die rechte Randbedingung einzuhalten. Umgekehrt werden die Positionskomponenten des Anfangspunkts korrigiert, wenn man die Geschwindigkeitskomponenten der Randpunkte als unveränderlich wählt. Im GUI lässt sich außerdem eine dritte Option einstellen, bei der alle Anfangsbedingungen korrigiert werden, mit dem Ziel, sowohl die Positions- als auch die Geschwindigkeitskomponenten des rechten Randvektors beizubehalten. Im letztgenannten Fall ist die Zielfunktion $\boldsymbol{f}$ daher die Differenz zwischen dem gegebenen Bahnvektor am rechten Rand und dem (ausgehend vom korrigierten Anfangsvektor mittels `orbint_field`) integrierten Bahnvektor. Für die anderen beiden Fälle besteht die Zielfunktion aus den entsprechenden Komponenten dieser Differenz. Die bereits in Abschnitt 2.3.4 beschriebenen Abbruchkriterien `TolFun` und `TolFun` sind ebenfalls im GUI einstellbar.

Im dritten Test wurde ein Zeitintervall (‚Schussweite') von 5 Sekunden gewählt in der Absicht, dass die resultierenden Orbits weiterhin möglichst gut zu den Messungen $\dot{\rho}$ passen. Dabei genügt es nicht, nur die Geschwindigkeitskomponenten der Randpunkte festzuhalten, weil die Gleichung (2.3) zur Berechnung von $\dot{\rho}$ ein Funktional des Geschwindigkeits- *und* Positionsvektors ist. Hier wurde deshalb der gesamte Bahnvektor am rechten Intervallrand als einzuhaltende Bedingung gewählt. Obwohl die Zeitintervalle zu den im ersten Test gewählten Teilintervallen identisch sind, verlaufen die Differenzen in Abb. 5.29 bzw. Abb. 5.30 keineswegs so glatt wie bei der kubischen Spline-Interpolation. Auch beim K-Band-Vergleich sind größere Differenzen $\Delta\dot{\rho}$ festzustellen (s. Abb. 5.31). Diese Ergebnisse sind folgendermaßen zu erklären: da der rechte, festzuhaltene Randpunkt eines Intervalls auch gleichsam der linke Randpunkt des nächsten Intervalls ist, bleibt dieser Punkt letztlich nicht unverändert, sondern wird ebenfalls korrigiert. Somit entstehen bei dieser Methode unvermeidlich Sprünge an den Intervallknoten. Um für die vorliegenden Orbits eine wirksamere Glättung zu erreichen, sind größere Intervalle, z.B. über 60 Sekunden, nötig. Einerseits wären dann in der Folge abermals die Abweichungen zu den K-Band-Messungen größer ($-30\,\mathrm{\mu m/s} < \Delta\dot{\rho} < 30\,\mathrm{\mu m/s}$ bei 60 Sek.), andererseits blieben auch alle gegebenen Zustandsvektoren innerhalb des Intervalls völlig unberücksichtigt, weil beim Einfach-Schießverfahren die Korrektur der Anfangsbedingungen nur mithilfe der Differenzen am rechten Intervall-Randpunkt geschätzt wird. Für längere Intervalle scheint die nächste Methode besser geeignet zu sein.

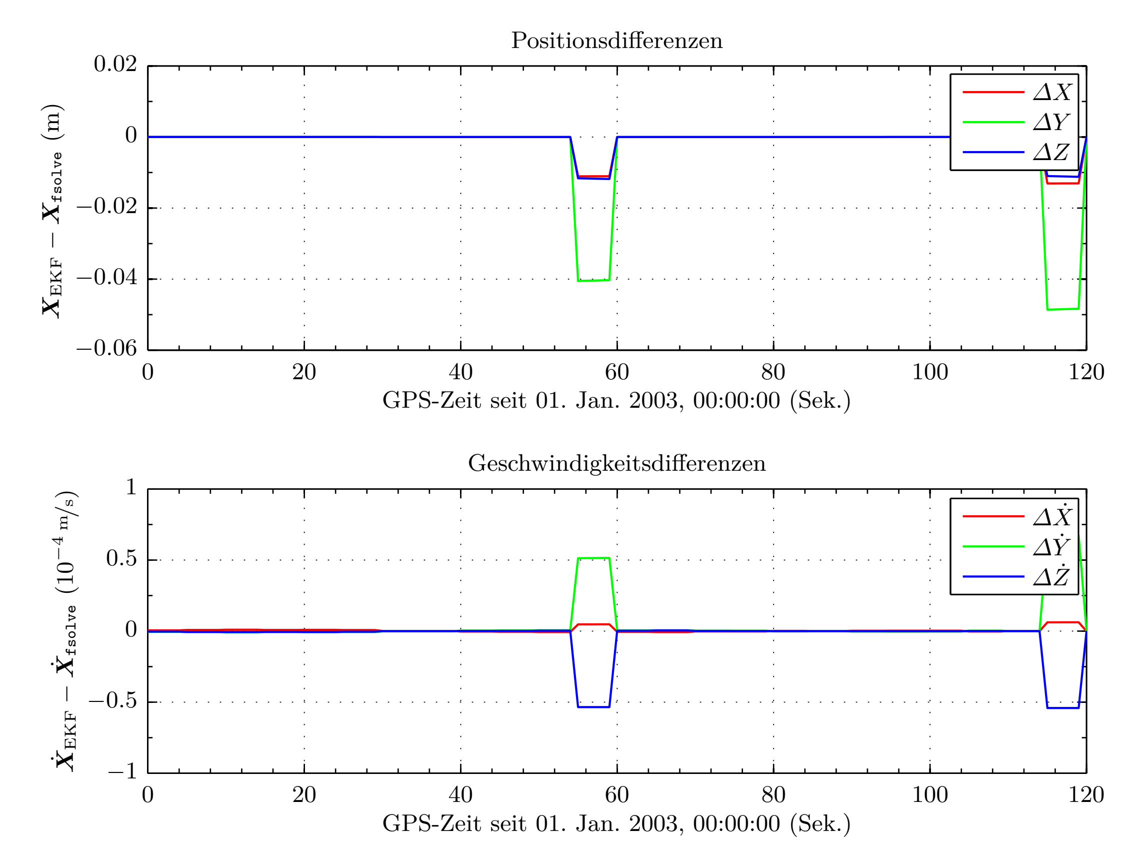


Abb. 5.29: Gefilterter vs. geglätteter GRACE-A-Orbit über 120 Sek. in Test 3

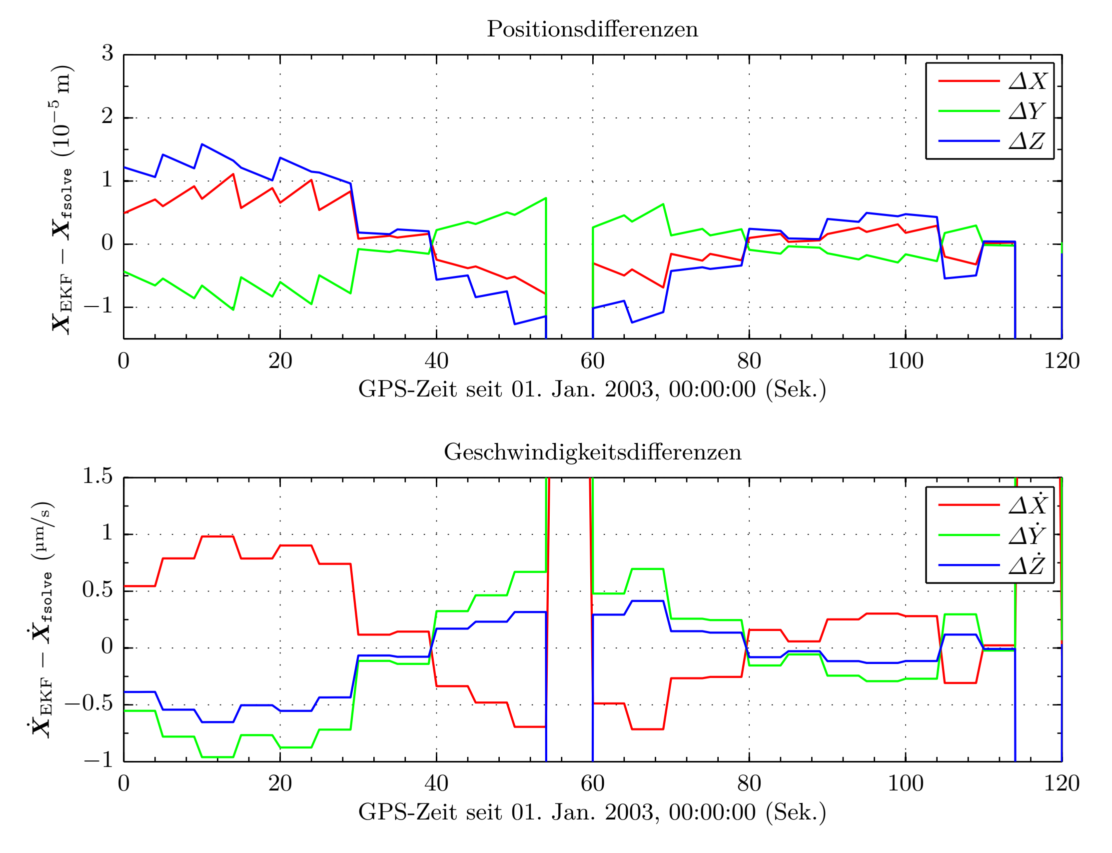


Abb. 5.30: Gefilterter vs. geglätteter GRACE-A-Orbit über 120 Sek. in Test 3 (vergrößerte Darstellung)

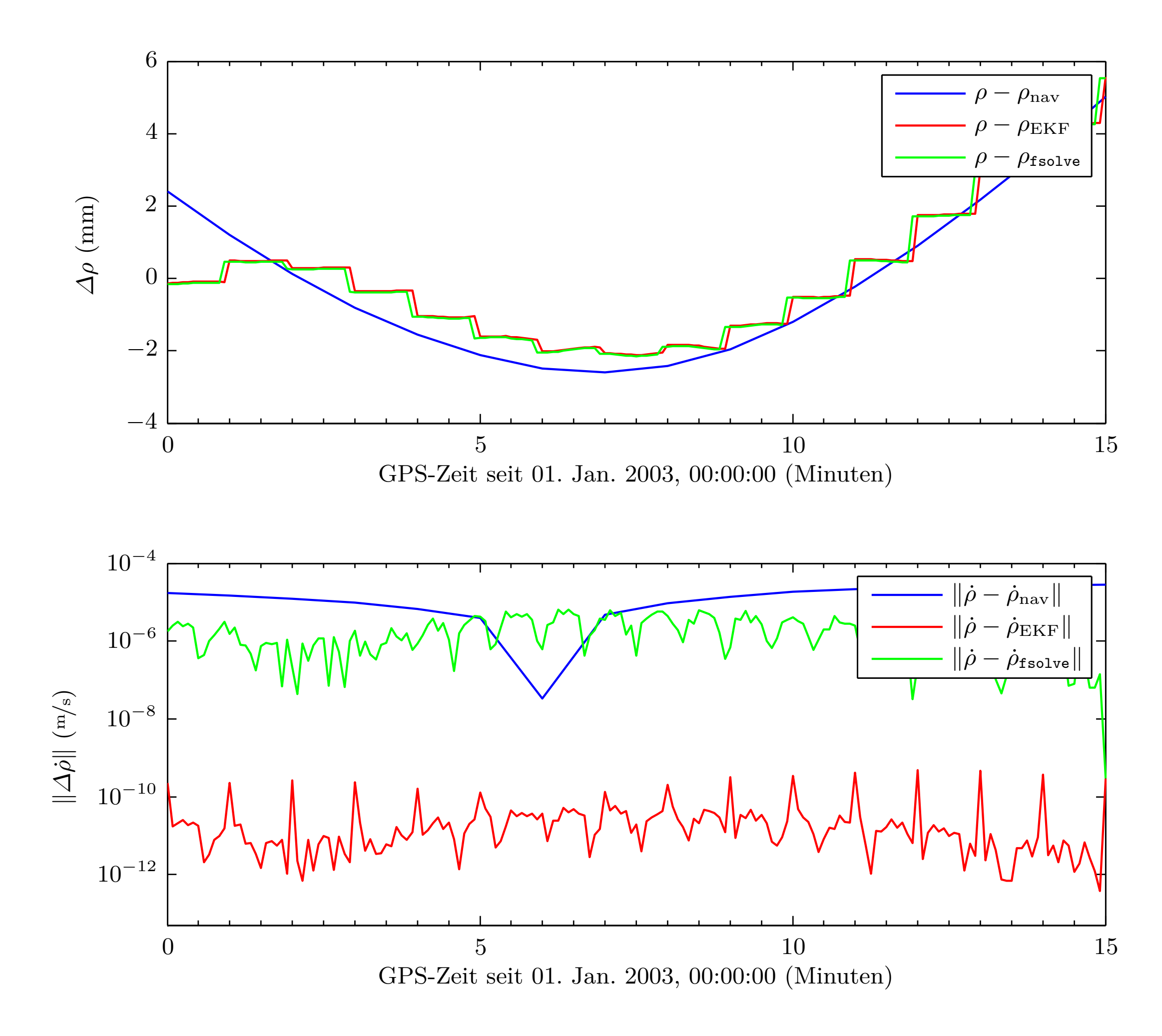


Abb. 5.31: K-Band-Vergleich zu Test 3

Methode 3 – Kleinste-Quadrate-Einpassung mittels `lsqnonlin`

Mit dem Schießverfahren hat die nun vorgestellte Glättungsmethode gemein, dass Korrekturen für die Anfangsbedingungen der Orbitintegration gesucht werden. Allerdings möchte man hier die Korrekturen nicht für die Lösung eines RWPs in dem Sinne nutzen, dass nach der Integration mit den korrigierten Anfangsbedingungen zwingend vorab festgelegte Randbedingungen am Intervallendpunkt eingehalten werden. Stattdessen sollen die Anfangswerte so gefunden werden, dass die Quadratsumme der Differenzen $\boldsymbol{f}$ minimal ist, wobei $\boldsymbol{f}$ hier die Abweichungen zwischen den gegebenen und den integrierten Zustandsvektoren sind (bzw. wahlweise nur die Positions- oder Geschwindigkeitskomponenten der Abweichungen). Somit kann letztlich mit dieser Methode erreicht werden, eine ausgleichende Bahnkurve zwischen ausgewählte Zustandsvektoren zu schätzen.

Während – allgemein betrachtet – mittels `fsolve` ein Vektor $\boldsymbol{x}$ gesucht wird, für den sich eine Nullstelle $\boldsymbol{f}(\boldsymbol{x}) \doteq 0$ ergibt, hat `lsqnonlin` gemäß O.T.-Handbuch (2010, S. 11-199) die Aufgabe, Probleme der Form

$$\min_{\boldsymbol{x}} \|\boldsymbol{f}\|_2^2 = \min_{\boldsymbol{x}} \big(f_1(\boldsymbol{x})^2 + f_2(\boldsymbol{x})^2 + \ldots + f_n(\boldsymbol{x})^2\big), \quad n \in \mathbb{N} \tag{5.1}$$

zu lösen, worin $\boldsymbol{x}$ ebenfalls ein Vektor und $\boldsymbol{f}(\boldsymbol{x})$ eine vektorwertige, in $\boldsymbol{x}$ nicht-lineare Funktion ist.

In `shootorb` erfolgt der Aufruf der Funktion `lsqnonlin` völlig analog zu `fsolve`, z.B. im Fall der Korrektur des gesamten Startvektors:

```
199 [corr,~,~,exitflag,output] = lsqnonlin(@shootfore6,[0 0 0 0 0 0],[],[],options);
200 ini_state = ini_state + corr;
```

Die Unterfunktion `shootfore6` liefert die Differenzen zwischen den integrierten und den gegebenen Bahnvektoren (an allen gemeinsamen Zeitpunkten) als Matrix zurück:

```
251 function F = shootfore6(corr)
252     [X] = orbint_field(field,tspan,ini_state+corr,type,Ri2e,lmax);
253     F(:,1:3) = X(ia,1:3) - pos(ib,3:5);
254     F(:,4:6) = X(ia,4:6) - vel(ib,3:5);
255 end
```

`lsqnonlin` überführt die Matrix `F` intern in einen Vektor und berechnet dann die Quadratsumme aus den Einträgen. Die Korrektur, für welche die minimale Quadratsumme folgt, wird wieder iterativ bestimmt, hier entsprechend der Standardeinstellung mithilfe des „trust-region-reflective“-Verfahrens (s. O.T.-Handbuch, 2010, S. 6-137 ff.). Alternativ ist die Lösung auch mittels des Levenberg-Marquardt- oder des Gauß-Newton-Algorithmus möglich, wozu man die Optionen mittels `optimset` anpassen müsste (s. O.T.-Handbuch, 2010, S. 11-212). Doch es besteht kein Anlass, einen anderen Algorithmus zu verwenden, da sich damit hier kein augenscheinlich besseres Ergebnis erreichen ließ.

Da im vierten Test ein Zeitintervall von 60 Sekunden gewählt wurde, zeigt der Kurvenverlauf der Differenzen $\Delta\rho$ beim K-Band-Vergleich in Abb. 5.32 (S. 129 links) in diesen Zeitabständen geringe Steigungsänderungen. Dass die Abweichungen $\Delta\dot{\rho}$ größer im Vergleich zu den gefilterten Orbits sind, ist nicht verwunderlich, denn die ausgleichenden Bahnkurven verlaufen nicht durch die in Zeitabständen von 5 Sekunden geschätzten Zustandsvektoren der gefilterten Orbits. Dies gilt auch für den fünften Test, bei dem die Korrektur der Anfangsbedingungen wegen des Inkrements $n = 1$ unter Einbeziehung *aller* innerhalb des 15-Minuten-Intervalls vorhandenen, gefilterten Bahnvektoren geschätzt wird. Für beide Satelliten wurde hierbei also jeweils nur eine ausgleichende Bahn berechnet. Infolgedessen sind beim K-Band-Vergleich in Abb. 5.33 (S. 129 rechts) kaum noch Unstetigkeiten auszumachen.

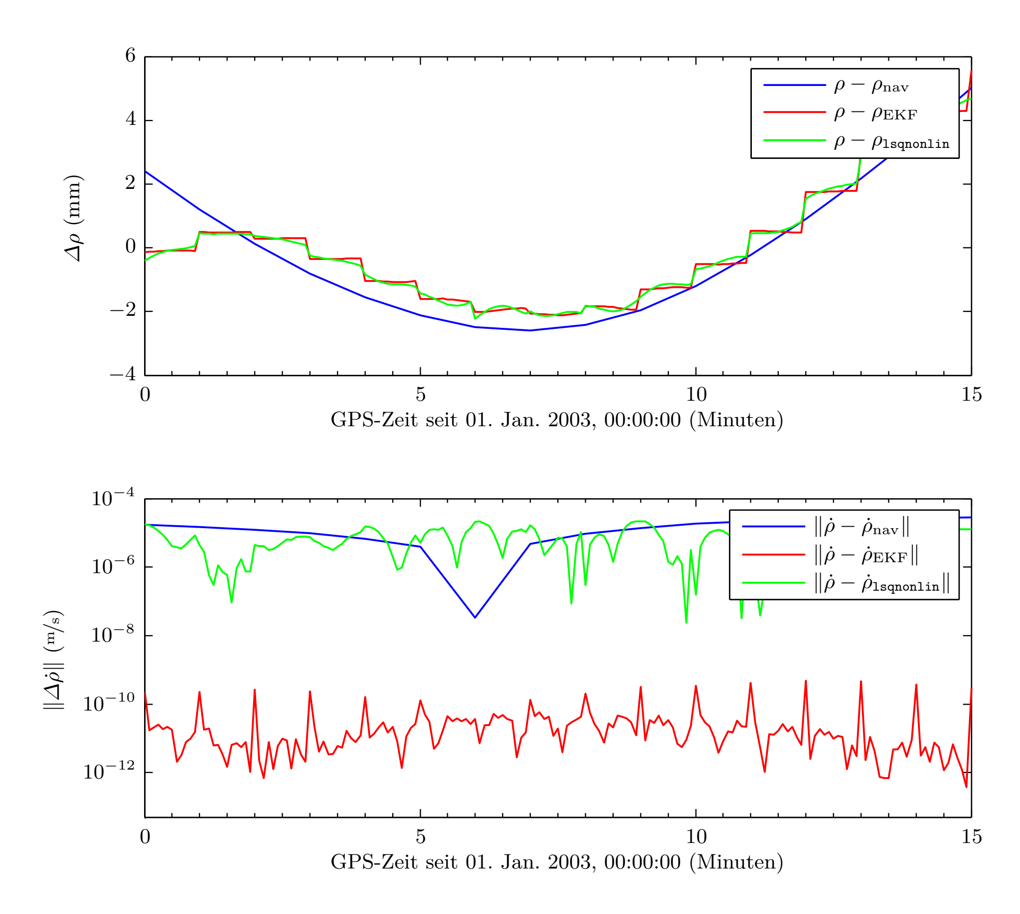


Abb. 5.32: K-Band-Vergleich zu Test 4

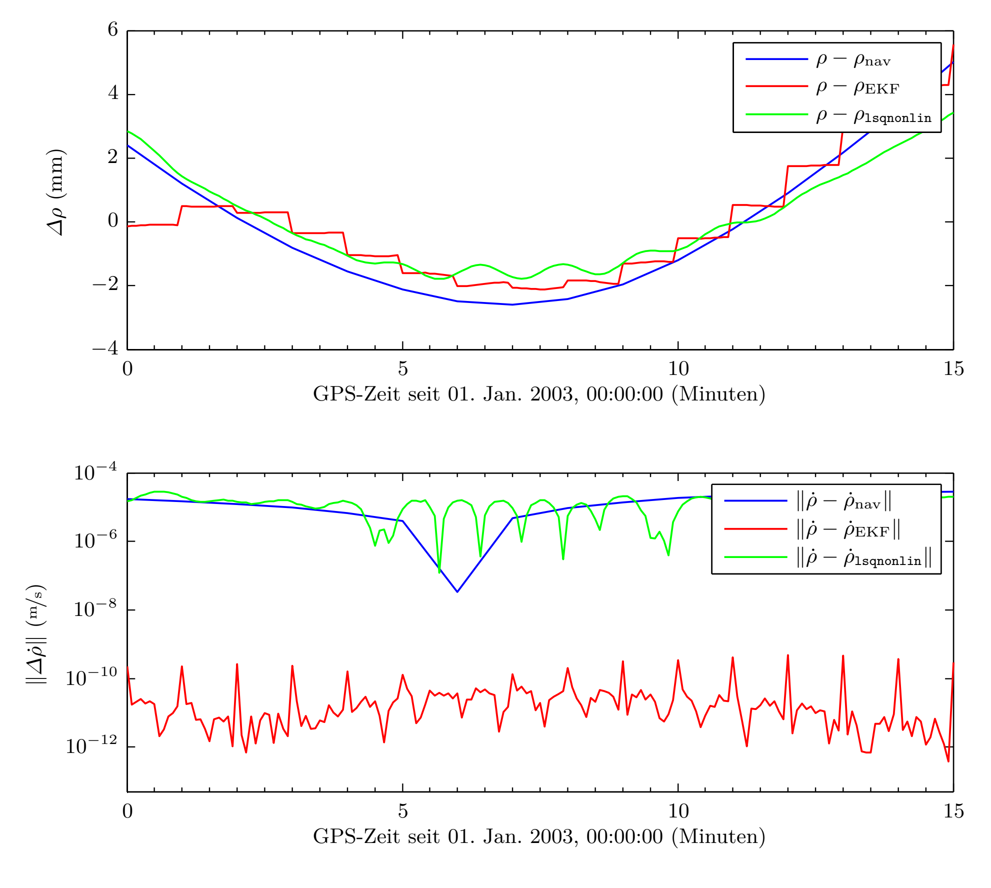


Abb. 5.33: K-Band-Vergleich zu Test 5

6 Kapitel 6
Schlussbetrachtungen und Ausblick

Diese Arbeit beabsichtigt einen Beitrag zu leisten zu möglichen Schwerefeldbestimmungen aus GRACE-Beobachtungen, indem sie sich mit dem Thema der Orbitverdichtung befasst, das ein Glied in der Prozesskette der Auswertungen zur Schwerefeldbestimmung darstellt.

Die hierfür eingangs gesetzten Ziele sind alle erreicht worden. Insbesondere ist es gelungen, eine MATLAB-Toolbox zu verwirklichen, mit der sich GRACE-Orbits dank der Kalman-Filterung derart verdichten lassen, dass sie außerordentlich gut zu den beobachteten K-Band-Relativgeschwindigkeiten passen.

Außerdem stellten sich in dieser Arbeit auch folgende zwei wichtige Erkenntnisse heraus:

- Bei der Fortpflanzung der Schätzfehlerkovarianzen mithilfe der Transitionsmatrix, die auf einem linearen Modell beruht, hat die Vernachlässigung von Entwicklungsgraden $l > 2$ in den Variationsgleichungen zumindest bei kurzen Integrationszeiten keine signifikante Auswirkung auf die prädizierten Kovarianzen (Abschnitt 4.6, S. 90 ff.). Insgesamt ist die Kovarianzprädiktion basierend auf dem linearen Modell mit dem bis Grad 2 ausgewerteten Marussi-Tensor in dieser Anwendung aufgrund der kurzen Integrationsintervalle von max. 60 Sekunden genügend genau.
- Auftretendes Messrauschen in den simulierten Orbits ließ sich mithilfe der Filter nur reduzieren, falls man zur Prädiktion der Systemzustände dasselbe Kräftemodell genutzt hat wie bereits zur Erzeugung des verrauschten Orbits (Abschnitt 4.7, S. 92 ff.).

Die verdichteten Orbits weisen kleinere[1] Sprünge auf, die methoden-immanent sind, weil sie durch die Arbeitsweise der erweiterten Kalman-Filterung in Verbindung mit der Orbitintegration zustande kommen. Um die Sprünge zu reduzieren bzw. zu eliminieren, wurden zusätzlich drei einfache Glättungsmethoden umgesetzt (s. Abschn. 5.3.6), welche, falls gewünscht, nach der Orbitverdichtung in der Toolbox angewendet werden können. Zusammenfassend lässt sich feststellen, dass jedoch keine dieser experimentellen Methoden die gewünschten optimalen Ergebnisse liefert, weil bei einer starken Glättung die hohe Übereinstimmung der Orbits mit den K-Band-Messungen verloren geht. Zum derzeitigen Stand der Toolbox muss man sich unter den folgenden Möglichkeiten bei der Glättung entscheiden:

- Möchte man die hohe Übereinstimmung der Orbits mit den K-Band-Relativgeschwindigkeiten $\dot{\rho}$ beibehalten, sollte man unter den experimentellen Methoden die kubische Spline-Interpolation mit der Option $n = 5$ wählen (wie in Test 1, siehe S. 122). Dabei muss man letztlich in Kauf nehmen, dass die wegen der Hauptkorrektur alle 60 Sekunden auftretenden Sprünge nur leicht abgemildert werden.
- Legt man hingegen Wert auf möglichst glatte Bahnkurven, ist die kleinste-Quadrate-Einpassung mittels `lsqnonlin` (s. Seite 128 ff.) zu favorisieren. Dabei muss man hinnehmen, dass die Orbits weniger gut zur K-Band-Beobachtung $\dot{\rho}$ passen.

1 Maximal 6.5 cm bei den Positions- und 0.2 mm/s bei den Geschwindigkeitskomponenten, s. Abschnitt 5.3.4, S. 113.

Möchte man das Hauptprogramm später mit den gegebenen oder anderen realen GRACE-Daten ausführen, wird neben dieser Entscheidung über die Glättungsmethoden außerdem unweigerlich die Frage aufkommen, welche Werte man am besten für die Parameter wählt, die die Gewichtung der K-Band-Messungen $\dot{\rho}$ steuern (also insb. $\sigma_{\dot{\rho}}$ und $\sigma_{0,\mathtt{navvel}}$). Da man vorhandene, reale GRACE-Navigationslösungen wahrscheinlich ohnehin nur dann mithilfe der erstellten Toolbox unter Verwendung der K-Band-Messungen verdichten wird, wenn man sie im Anschluss zu einer Schwerefeldbestimmung nutzen möchte, wäre als Entscheidungshilfe vorstellbar, die Auswirkung einer mehr oder weniger starken K-Band-Anpassung auf das Ergebnis dieser anschließenden Auswertung zu untersuchen. Da z.B. beim SGG-Ansatz zur Schwerefeldbestimmung zwei GRACE-Orbits mit den K-Band-Beobachtungen kombiniert werden, um die gesuchte Schwerefeldinformation zu bestimmen, erhofft man sich, dass dabei eine Genauigkeitssteigerung der Schwerefeldbestimmung eintritt.

Ob sich also die so verdichteten GRACE-Orbits tatsächlich bei der weiteren Schwerefeldauswertung als nützlich erweisen, müssen zukünftige Untersuchungen zeigen. Möglicherweise wird dabei der Genauigkeitsgewinn der Schwerefeldlösung nicht so groß ausfallen wie erhofft. Denn die hier verdichteten Orbits wurden nur mithilfe der Relativgeschwindigkeit $\dot{\rho}$ optimiert. Die für die Qualität der Schwerefeldlösung nach dem SGG-Ansatz ausschlaggebende Größe $\|\dot{\boldsymbol{X}}_{12}\|$ hängt jedoch nicht allein von der Relativgeschwindigkeit $\dot{\rho}$ ab, wie anhand des Zusammenhangs $\dot{\rho} = \langle \dot{\boldsymbol{X}}_{12}, \boldsymbol{e}_{12} \rangle$ aus Gl. (2.3) erkennbar ist. Auch muss $\|\dot{\boldsymbol{X}}_{12}\|$ nicht notwendigerweise in derselben räumlichen Richtung liegen wie die K-Band-Messungen, welche stets auf die LOS-Richtung bezogen sind (vgl. Abb. 2.2, S. 8). Denn zufolge Case et al. (2010, S. 4) sind die KBR1B-Daten im satellitenfesten Referenzsystem SRF („science reference frame“) gegeben, dessen Achse $\boldsymbol{x}_{\text{SRF}}$ gerade in die LOS-Richtung zeigt.

Im Hinblick auf zukünftige Studien wäre vorstellbar, die umgesetzte Orbitverdichtung für $\|\dot{\boldsymbol{X}}_{12}\|$ dadurch weiter zu optimieren, indem man zusätzlich zu den K-Band-Daten KBR1B auch die Level-1B-Sternkameradaten („Star Camera Assembly Data Format Record“ SCA1B) verwendet. Diese verzeichnen laut Wu et al. (2006, Abschn. 4.1.3, S. 17) Quaternionen zur Rotation vom raumfesten Referenzsystem ICRF in das SRF, ebenfalls mit einer zeitlichen Auflösung von 5 Sekunden. Da $\boldsymbol{x}_{\text{SRF}}$ in die LOS-Richtung weist, ist der LOS-Einheitsvektor $\boldsymbol{e}_{12}$ im SRF einfach mit $[1,\ 0,\ 0]^\top$ für alle Zeitpunkte anzunehmen. Folglich müsste sich eine Pseudobeobachtung für $\boldsymbol{e}_{12}$ im raumfesten System durch eine Rücktransformation mithilfe der SCA1B-Quaternionen bestimmen lassen. Diese könnte dann zusammen mit geeignet abgeleiteten Standardabweichungen als weitere Beobachtung in die Kalman-Filterung eingehen, in analoger Weise wie hier bereits $\dot{\rho}$. Alternativ könnte man mithilfe der Pseudobeobachtung für $\boldsymbol{e}_{12}$ aus Gl. (2.3) auch unmittelbar für $\dot{\boldsymbol{X}}_{12}$ eine Beobachtung mit zugehörigen Kovarianzen herleiten und damit entsprechend die prädizierten Systemzustände aufdatieren. In beiden Fällen ist es allerdings angebracht, vorher zu prüfen, ob die neu ermittelten Pseudomessungen hinreichend zuverlässig und genau sind, z.B. durch einen Vergleich mit den entsprechenden, aus den Navigationslösungen abgeleiteten Größen.

Abgesehen von dieser Maßnahme, zusätzliche Beobachtungen einzubeziehen, können noch folgende Empfehlungen bzw. Verbesserungsmöglichkeiten für zukünftige Arbeiten genannt werden:

- *Erweiterung des deterministischen Anteils des Systemmodells*, d.h. der Kräftefunktion, welche numerisch integriert wird. Dabei sollte man mit den größten, bislang nicht berücksichtigten Störeinflüssen beginnen. Da GRACE zu den LEO-Satelliten zählt, sind dies hier sowohl die Atmosphärenreibung als auch die direkte gravitative Wirkung von Drittkörpern (Mond und Sonne) und die indirekten Gezeiteneinflüsse (Ozeangezeiten und Gezeiten der festen Erde). Sofern man die Verdichtung im Hinblick auf GNV1B-Orbits optimieren möchte, hierbei also kleinere Korrekturen bei den Haupt-Messungsupdates wünscht, sollten bei der Ergänzung der Kräftefunktion die in Abschnitt 5.1, S. 99 erwähnten Modelle zur Anwendung kommen, da vom JPL wohl diese Modelle zur Erstellung der GNV1B-Orbits genutzt werden.

- *Überarbeitung des stochastischen Modells für das Systemrauschen.* Die getroffene Annahme weißen Rauschens, welche damit einhergeht, dass in der Matrix `Q` nur die Diagonalenelemente besetzt sind, sollte revidiert werden, denn streng gesehen ist es nicht realistisch, diesen

Rauschprozess zu unterstellen, s. Abschnitt 3.2.4 (S. 39 ff.). Stattdessen könnte man etwa den dort angeführten Ansatz von Wright et al. (2008) oder einen einfacheren, nicht-physikalisch motivierten *farbigen* Rauschprozess erwägen (z.B. Simon, 2006, Abschn. 7.1). Ferner könnte die bereits auf Seite 28 erwähnte Möglichkeit, eine Korrelation zwischen System- und Messrauschen zuzulassen, vorteilhaft sein (s. Simon, 2006, Abschn. 7.2).

- *Entwurf und Implementierung eines alternativen Filters*, welches die nicht-linearen Beobachtungs- und Systemgleichungen zur Einbringung bzw. Prädiktion der Kovarianzen besser approximiert. Eigentlich würden nur echte nicht-lineare Filter optimale Schätzwerte (i. S. der Bayes'schen Schätztheorie) bei nicht-linearen Problemstellungen liefern. Die Realisierung eines solchen Filters ist unter Umständen sehr aufwendig oder gar unmöglich, u.a. müsste zur nicht-linearen Fortschreibung der Kovarianzmatrizen des Zustandsvektors eine je nach Systemmodell spezifische, nicht-lineare partielle DGL (sog. Fokker-Planck- oder Kolmogorov-Gleichung) numerisch gelöst werden (vgl. Maybeck, 1982a, S. 192). Allerdings ist es durchaus machbar, höhere Grade bei den Taylorreihenentwicklungen der nicht-linearen System- und Beobachtungsfunktionen zu berücksichtigen. Bricht man also Gleichung (3.28) für die Systemfunktion $\boldsymbol{f}$ und Gl. (3.29) für die Messfunktion $\boldsymbol{h}$ z.B. erst nach dem zweiten Grad ab, könnte man auf diese Weise ein EKF zweiter Ordnung herleiten[1].

  In Boden (1986) lieferte ein solches Gauß-Filter zweiter Ordnung bei der Orbitbestimmung für alle Satellitenszenarien gegenüber des EKFs erster Ordnung und der sequentiellen Ausgleichung durchweg bessere Schätzergebnisse. Boden (1986, S. 76/77) stellt fest, dass bei einer Orbitbestimmung der Term zweiter Ordnung in der Reihenentwicklung der Beobachtungsgleichung $\boldsymbol{h}$ signifikanter sei als der Term zweiter Ordnung der approximierten Systemdynamik-Funktion $\boldsymbol{f}$, d.h. die Nichtlinearitäten der Messgleichungen haben einen größeren Effekt als die der Systemgleichung.

  Zumindest, solange das Systemrauschen noch als simpler weißer Rauschprozess angenommen wird, lohnt es sich also nicht unbedingt, Terme zweiter oder höherer Ordnung bei der Kovarianzprädiktion einzukalkulieren. Hingegen sollte man beim Korrekturschritt in Betracht ziehen, die Filtergleichungen eines Filters höherer Ordnung anzusetzen, da offensichtlich bei nicht-linearen Messgleichungen (hier für $\dot{\rho}$ und evtl. später $\boldsymbol{e}_{12}$) die Terme höherer Ordnung durchaus signifikant sein können.

  Freilich könnte auch untersucht werden, ob andere, für nicht-lineare Probleme ausgelegte Filtertypen bessere Schätzergebnisse bei der Orbitverdichtung liefern als z.B. ein Gauß-Filter zweiter Ordnung. Die Herleitung eines solchen Filters beginnt oftmals damit, indem die Taylorreihenentwicklung der nicht-linearen Systemfunktionen $\boldsymbol{h}$ bzw. $\boldsymbol{f}$ beim EKF durch eine andere Approximation ersetzt wird, etwa durch zentrale Differenzenquotienten, Gauß-Hermite-Quadraturformeln oder durch statistische Näherungen, z.B. die sog. „unscented transform“, welche auf das UKF („Unscented Kalman filter“) führt (Näheres z.B. in Grewal und Andrews, 2008, Abschn. 7.3.3, S. 334 ff.; Särkkä, 2006, S. 65 ff.; Simon, 2006, Kap. 14, S. 433 ff.). Sowohl mittels eines Gauß-Filters zweiter Ordnung als auch mithilfe eines UKFs ließen sich höchstwahrscheinlich hier gegenüber dem EKF die Schätzergebnisse etwas verbessern.

- *Umsetzung der Orbitverdichtung mittels eines Glättungsfilters.* Dieses nur im Postprocessing verwendbare Filter vermag zuverlässigere und genauere Schätzungen der Systemzustände zu liefern[2]. Außerdem könnten eventuell die unerwünschten Sprünge im Systemzustand im Vergleich zur Kalman-Filterung gemindert werden (hierauf wurde bereits in Abschnitt 3.2.7 hingewiesen).

1 Auch als Gauß-Filter zweiter Ordnung bezeichnet, da in der Formel für die Kalman-Matrix i.d.R. eine Gauß'sche Näherung nötig wird, vgl. Gelb (1996, S. 192).

2 Siehe z.B. Grewal und Andrews (2008, Abschn. 5.1.6: Improvement Over Filtering, S. 187 ff.) oder Simon (2006, Abschn. 9.2.1: Estimation improvement due to smoothing, S. 270 ff.).

Um die hier erstellte Toolbox zur Orbitverdichtung diesbezüglich auszubauen, wäre beispielsweise ein (kontinuierlich-diskretes) erweitertes Glättungsfilter nach RAUCH ET AL. (1965) geeignet, wie es u.a. in HARTIKAINEN und SÄRKKÄ (2008, Abschn. 2.2.3, S. 16)[1] und SÄRKKÄ (2006, Algorithm 3.2.3, S. 95) beschrieben wird. Den benötigten Formelapparat findet man in ähnlicher Form auch in GELB (1996, Tab. 5.2-2, S. 164). Da dieses Glättungsfilter wie das EKF die Taylor-Approximation nutzt, existieren wiederum Glättungsfiltervarianten, die höhere Ordnungen berücksichtigen. Der dabei wie auch beim EKF höherer Ordnung in Aussicht gestellten Zunahme der Schätzgenauigkeiten steht gegenüber, dass zunächst die Umsetzung und später auch die Berechnungen selbst mit erheblich mehr Aufwand verbunden sind. Dies ist im Falle des UKF nicht anders, zu dem es übrigens ebenfalls ein passendes Glättungsfilter gibt („unscented Kalman smoother“, s. z.B. SÄRKKÄ, 2006, S. 69 ff.).

- *Effizienzsteigerung der Berechnungen in MATLAB:*

  Bei den Integratoren `orbint_field` und `orbint_field2` wird (zum derzeitigen Stand) im zeitdiskreten Fall `'var'` die Transitionsmatrix $\boldsymbol{\Phi}$ durch numerische Integration bestimmt. Um die Berechnungszeit zu verkürzen, könnte man andere Methoden zur Bestimmung von $\boldsymbol{\Phi}$ testen. Womöglich genügt bereits die analytische Näherung basierend auf der Reihenentwicklung der Matrixexponentialfunktion entsprechend Gleichung (3.47). Diese Vereinfachung ist u.a. beim diskreten EKF möglich, nicht z.B. beim UKF, dessen Filtergleichungen die Matrix $\boldsymbol{\Phi}$ nicht enthalten.

  Anwenden der Parallel Computing Toolbox™, um das Potential eines Computers, der mit einem Mehrkernprozessor bzw. einer speziellen leistungsfähigen GPU ausgestattet ist, voll ausnutzen zu können[2]. Dadurch ließe sich die Berechnungsgeschwindigkeit steigern, ohne dass eine Vereinfachung der Berechnungen nötig wäre.

Die Wissenschaft braucht Zusammenarbeit, in der sich das Wissen des einen durch die Entdeckungen des andern bereichert.

(JOSÉ ORTEGA Y GASSET, 1883-1955)

1 Dokumentation zur EKF/UKF-Toolbox von HARTIKAINEN und SÄRKKÄ. In deren MATLAB-Toolbox ist die Funktion `erts_smooth1` für diese Glättung zuständig.

2 http://www.mathworks.de/products/parallel-computing/

A

Anhang A

MATLAB-Quellcodes

Nachfolgend sind Informationen zu den Programmen und zu einigen Funktionen der umgesetzten Toolbox (Verzeichnis `\ekf_orbit_toolbox` auf der beiliegenden CD) gegeben.

A.1 Programme

Den sechs in der Toolbox enthaltenen Hauptprogrammen obliegen nachstehende Aufgaben:

- `ekf_orbit_main`: Verdichtung realer GRACE-Daten mittels EKF, Nachprozessierung durch einfache Glättungsmethoden sowie Analyse, Darstellung und Sicherung der Ergebnisse. Die gesamte Bedienung erfolgt über GUIs, wie es schon in Abschnitt 4.1 beschrieben wurde.
- `ekf_test_main`: Test der implementierten EKF-Orbitverdichtung mithilfe simulierter Daten. Mit dem Programm wird nachgewiesen, dass die gefilterten bzw. verdichteten Orbits geringere Abweichungen zu den Referenzorbits aufweisen als die simulierte Navigationslösung. Dies bedeutet, dass die umgesetzten Filtermechanismen wirksam funktionieren (vgl. Abschnitt 4.7).
- `Grace_analysis_main`: Überprüfung der realen GRACE-Daten, *bevor* sie mit `ekf_orbit_main` weiterverarbeitet werden. Dabei vergleicht man in der Hauptsache die gegebene Navigationslösung mit den K-Band-Beobachtungen (siehe Abschnitt 5.2).
- `orbint_test_main`: Test der Orbitintegratoren u. der Fehlerfortpflanzung (Abschn. 4.5 u. 4.6).
- `simple_odm_sim_main` und `simple_odm_real_main`: Verdichtung simulierter bzw. realer Orbits mit einfachen Interpolations- und Integrationsverfahren (Abschnitt 2.3).

A.2 Funktionen

Hier wird nun für ausgewählte Funktionen dargelegt, welchem Zweck sie dienen und über welche Ein- und Ausgabeargumente sie verfügen. Eine Übersicht *sämtlicher* Funktionen gibt die Textdatei `contents.txt`. Dort ist auch eine kurze Beschreibung zu jeder Funktion gegeben.

Funktion A.1: `cart2kepler`

Aufgabe	Umrechnung eines quasi-inertialen Positions- und Geschwindigkeitsvektors in Keplersche Bahnelemente.	
Eingangsargumente		
`r_i`	(3×1)	Positionsvektor $\boldsymbol{X}$ im geozentrisch gelagerten, raumfesten Bezugssystem
`r_i_p`	(3×1)	Geschwindigkeitsvektor $\dot{\boldsymbol{X}}$ in ebendiesem System
`type`	string/struct	zu verwendende Konstanten
Ausgabeargumente		
`a`	(1×1)	große Halbachse a
`e`	(1×1)	Exzentrizität e
`I`	(1×1)	Inklination i
`omega`	(1×1)	Argument des Perigäums ω
`Omega`	(1×1)	Rektaszension des aufsteigenden Knotens $\varOmega$
`M`	(1×1)	mittlere Anomalie M
`u`	(1×1)	Argument der Breite u
`nu`	(1×1)	wahre Anomalie ν
`E`	(1×1)	exzentrische Anomalie E

Funktion A.2: `ekf_orbit`

Aufgabe	Verdichtung quasi-inertialer Satellitenorbits mittels EKF (*ohne* Nutzung von K-Band-Messungen) durch geeigneten Aufruf der Unterfunktionen `ekf_predict` und `ekf_update`	
Eingangsargumente		
`navpos_i`	$(n \times 11)$	Zeitreihe der zu verdichtenden Positionen, Matrixstruktur s. Abschn. 4.1.1
`navvel_i`	$(n \times 11)$	Zeitreihe der zu verdichtenden Geschwindigkeiten
`userstoptime`	(1×1)	gewünschter letzter Zeitpunkt des verdichteten Orbits
`Q`	(6×6)	Spektraldichte- od. Kovarianzmatrix des Systemrauschens (abh. von `cov_sel`)
`type`	string/struct	zu verwendende Konstanten
`cov_sel`	string	Wahl der Kovarianzfortschreibungsmethode (`'ric'`, `'lin'` od. `'const'`)
`orbint_sel`	string	Wahl des Orbitintegrators (`'kep'`, `'kep2'`, `'j2'`, `'field'` od. `'field2'`)
`wb_sel`	boolean	Fortschrittsanzeige („wait bar“) aktivieren/deaktivieren
`Ri2e`	$(r \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$, Matrixstruktur s. Abschn. 4.1.1
`fs_R`	(1×1)	Sampling der Zeitreihe `Ri2e`
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`lmax`	(1×1)	Maximalgrad bei Reihenentwicklung für $\nabla_h V$
`lmax2`	(1×1)	Maximalgrad bei Berechnung des Marussi-Tensors $\boldsymbol{M}_h$ (empfohlen < 12)
`tens_sel`	string	Falls `'PetVer'`, Berechnung von $\boldsymbol{M}_e$ durch `tenspshs`, ansonsten durch `tenssp`
Ausgabeargumente		
`intpos`	$(m \times 11)$	Zeitreihe der verdichteten Positionen (Matrixstruktur analog zu `navpos_i`)
`intvel`	$(m \times 11)$	Zeitreihe der verdichteten Geschwindigkeiten (1-Sek.-Sampling)
`diff_X_hat_X`	$(n \times 8)$	bei Messungsupdates geschätzte Korrekturen an den Systemzustand $\boldsymbol{\delta\hat{s}}$. Spalten: [MJD t $\varDelta X$ $\varDelta Y$ $\varDelta Z$ $\varDelta\dot{X}$ $\varDelta\dot{Y}$ $\varDelta\dot{Z}$]
`X_end`	$(n \times 8)$	prädizierte/integrierte Systemzustände $\bar{\boldsymbol{s}}$ an Messungsupdates. Spalten: [MJD t X Y Z $\dot{X}$ $\dot{Y}$ $\dot{Z}$]
`P_end`	$(6 \times 6 \times n)$	Kovarianzmatrizen $\bar{\boldsymbol{P}}$ der prädizierten Zustände $\bar{\boldsymbol{s}}$
`calctime`	(1×1)	verstrichene Berechnungszeit (in Sek.)

Funktion A.3: `ekf_orbit2`

Aufgabe	Verdichtung quasi-inertialer GRACE-Orbits mittels EKF (*mit* Nutzung der K-Band-Messungen $\dot{\rho}$) durch geeigneten Aufruf der Unterfunktionen `ekf_predict` und `ekf_update`	
Eingangsargumente		

Fortsetzung `ekf_orbit2`

`navposA_i`	$(n \times 11)$	Zeitreihe der Positionskomponenten der GRACE-A-Navigationslösung, Matrixstruktur s. Abschn. 4.1.1
`navvelA_i`	$(n \times 11)$	Zeitreihe der Geschwindigkeitskomponenten der GRACE-A-Navigationslösung
`navposB_i`	$(n \times 11)$	Zeitreihe der Positionskomponenten der GRACE-B-Navigationslösung
`navvelB_i`	$(n \times 11)$	Zeitreihe der Geschwindigkeitskomponenten der GRACE-B-Navigationslösung
`kband`	$(w \times 5)$	Zeitreihe der K-Band-Messungen. Matrixstruktur s. Abschn. 4.1.1
`userstoptime`	(1×1)	gewünschter letzter Zeitpunkt des verdichteten Orbits
`Q`	(6×6)	Spektraldichte- od. Kovarianzmatrix des Systemrauschens (abh. von `cov_sel`)
`type`	string/struct	zu verwendende Konstanten
`cov_sel`	string	Wahl der Kovarianzfortschreibungsmethode (`'ric'`, `'lin'` od. `'const'`)
`orbint_sel`	string	Wahl des Orbitintegrators (`'kep'`, `'kep2'`, `'j2'`, `'field'` od. `'field2'`)
`wb_sel`	boolean	Fortschrittsanzeige („wait bar“) aktivieren/deaktivieren
`std_rhop`	(1×1)	Standardabweichung der gemessenen Relativgeschwindigkeiten $\dot{\rho}$
`sigma0_vel`	(1×1)	$\sigma_{0,\texttt{navvel}}$ gewichtet Geschwindigkeiten der Navigationslösung bei Updates herab
`Ri2eA`	$(r \times 9)$	Zeitr. d. Rotationsmatrizen $\boldsymbol{R}_e^i$ für GRACE-A, Matrixstruktur s. Abschn. 4.1.1
`Ri2eB`	$(r \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$ für GRACE-B
`fs_R`	(1×1)	Sampling der Zeitreihe `Ri2e`
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`lmax`	(1×1)	Maximalgrad bei der Reihenentwicklung für $\nabla_h V$
`lmax2`	(1×1)	Maximalgrad bei der Berechnung des Marussi-Tensors $\boldsymbol{M}_h$ (empfohlen < 12)
`tens_sel`	string	Falls `'PetVer'`, Berechnung von $\boldsymbol{M}_e$ durch `tenspshs`, ansonsten durch `tenssp`
Ausgabeargumente		
`intposA`	$(m \times 11)$	Zeitreihe der verdichteten GRACE-A-Positionen (Spalten analog zu `navpos_i`)
`intvelA`	$(m \times 11)$	Zeitreihe der verdichteten GRACE-A-Geschwindigkeiten (1-Sek.-Sampling)
`intposB`	$(m \times 11)$	Zeitreihe der verdichteten GRACE-B-Positionen
`intvelB`	$(m \times 11)$	Zeitreihe der verdichteten GRACE-B-Geschwindigkeiten
`diff_X_hat_X`	$(n \times 14)$	bei Messungsupdates geschätzte Korrekturen an den Systemzustand $\boldsymbol{\delta\hat{s}}$. Spalten: [MJD t ΔX_1 … $\Delta \dot{Z}_1$ ΔX_2 … $\Delta \dot{Z}_2$]
`X_end`	$(n \times 14)$	prädizierte/integrierte Systemzustände $\bar{\boldsymbol{s}}$ an Messungsupdates. Spalten: [MJD t X_1 Y_1 Z_1 $\dot{X}_1$ $\dot{Y}_1$ $\dot{Z}_1$ X_2 Y_2 Z_2 $\dot{X}_2$ $\dot{Y}_2$ $\dot{Z}_2$]
`P_end`	$(12 \times 12 \times n)$	Kovarianzmatrizen $\bar{\boldsymbol{P}}$ der prädizierten Zustände $\bar{\boldsymbol{s}}$
`calctime`	(1×1)	verstrichene Berechnungszeit (in Sek.)

Funktion A.4: `ekf_predict`

Aufgabe	Durchführung *eines* Prädiktionsschritts für die Funktionen `ekf_orbit` und `ekf_orbit2` mithilfe des gewählten Orbitintegrators ausgehend von Startwerten $\hat{\boldsymbol{s}}$, $\widehat{\boldsymbol{P}}$	
Eingangsargumente		
`X_hat`	(1×6)	Startvektor des aktuellen Prädiktionsschritts (zuvor geschätzter Zustand $\hat{\boldsymbol{s}}$)
`P_hat`	(6×6)	zugehörige Kovarianzmatrix $\widehat{\boldsymbol{P}}$
`tspan`	$(n \times 1)$	Vektor der Prädiktionszeitpunkte, hier: $[0, 1, \ldots, 60]$ bzw. $[0, 1, \ldots, 5]$ Sek.
`Q`	(6×6)	Spektraldichte- od. Kovarianzmatrix des Systemrauschens (abh. von `cov_sel`)
`type`	string/struct	zu verwendende Konstanten
`cov_sel`	string	Wahl der Kovarianzfortschreibungsmethode (`'ric'`, `'lin'` od. `'const'`)
`orbint_sel`	string	Wahl des Orbitintegrators (`'kep'`, `'kep2'`, `'j2'`, `'field'` od. `'field2'`)
`Ri2e`	$(r \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$, Matrixstruktur s. Abschn. 4.1.1
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`lmax`	(1×1)	Maximalgrad bei der Reihenentwicklung für $\nabla_h V$
`lmax2`	(1×1)	Maximalgrad bei der Berechnung des Marussi-Tensors $\boldsymbol{M}_h$ (empfohlen < 12)
`tens_sel`	string	Falls `'PetVer'`, Berechnung von $\boldsymbol{M}_e$ durch `tenspshs`, ansonsten durch `tenssp`
Ausgabeargumente		
`X`	$(n \times 6)$	Zeitreihe der prädizierten Zustandsvektoren $\bar{\boldsymbol{s}}$
`P`	$(n \times 36)$	Zeitreihe der propagierten Kovarianzmatrizen $\bar{\boldsymbol{P}}$ (im Zeilenformat)

Funktion A.5: `ekf_update`

Aufgabe	Durchführung *eines* Korrekturschritts beim EKF (universell anwendbar)	
Eingangsargumente		
`X`	$(u \times 1)$	prädizierter Systemzustand $\bar{\boldsymbol{s}}$
`P`	$(u \times u)$	prädizierte Kovarianzmatrix $\overline{\boldsymbol{P}}$
`z`	$(n \times 1)$	Beobachtungsvektor $\boldsymbol{b}$
`R`	$(n \times n)$	Kovarianzmatrix der Beobachtungen $\boldsymbol{R}$
`H`	$(n \times u)$	Messmatrix $\boldsymbol{H} = \frac{\partial \boldsymbol{h}(\boldsymbol{s})}{\partial \boldsymbol{s}}\big\|_{\bar{\boldsymbol{s}}}$ (erste partielle Ableitungen der Messgleichung $\boldsymbol{h}$ nach $\boldsymbol{s}$, ausgewertet an der Stelle $\bar{\boldsymbol{s}}$)
`h`	$(u \times 1)$	prädizierte Beobachtung $\boldsymbol{h}(\bar{\boldsymbol{s}})$ (Messgleichung $\boldsymbol{h}$ ausgewertet an der Stelle $\bar{\boldsymbol{s}}$)
Ausgabeargumente		
`X_hat`	$(u \times 1)$	korrigierter Systemzustand $\hat{\boldsymbol{s}}$
`P_hat`	$(u \times u)$	zugehörige Kovarianzmatrix $\widehat{\boldsymbol{P}}$
`corr`	$(u \times 1)$	geschätzte Korrektur an prädizierten Systemzustand $\boldsymbol{\delta}\hat{\boldsymbol{s}}$
`K`	$(u \times n)$	Kalman-Verstärkungs-/Gewichtsmatrix $\boldsymbol{K}$

Funktion A.6: `fgkepler`

Aufgabe	Berechnung eines zukünftigen Zustandsvektors $\bar{\boldsymbol{s}}$ und der zugehörigen Fehlertransitionsmatrix $\boldsymbol{\Phi}$ nach der analytischen Lagrange/Gibbs-Lösung des Zweikörperproblems (verwendet in `orbint_kep2`).	
Eingangsargumente		
`X0`	(1×6)	Startvektor $\hat{\boldsymbol{s}}_0$ (Anfangsposition und -geschwindigkeit im quasi-inertialen Bezugssystem)
`Delta_t`	(1×1)	Zeitdifferenz $t - t_0$ (Prädiktionszeitpunkt abzüglich Startzeitpunkt).
`a`	(1×1)	große Halbachse a
`e`	(1×1)	Exzentrizität e
`E0`	(1×1)	exzentrische Anomalie E_0 am Startzeitpunkt t_0
`E`	(1×1)	exzentrische Anomalie E am Prädiktionszeitpunkt t
`type`	string/struct	zu verwendende Konstanten (GM)
`outinfo`	string	Falls `'num'`, erfolgt Ausgabe in numerischer anstatt symbolischer Darstellung
Ausgabeargumente		
`X`	(1×6)	prädizierter Bahnvektor $\bar{\boldsymbol{s}}$
`Phi`	(1×36)	zugehörige Transitionsmatrix $\boldsymbol{\Phi}$

Funktion A.7: `grace_parms`

Aufgabe	Berechnung der K-Band-Größen ρ, $\dot{\rho}$ und des Einheitsvektors $\boldsymbol{e}_{12}$ aus quasi-inertialen Positions- und Geschwindigkeitsvektoren zweier Satellitenorbits gemäß Gl. (2.1) bis (2.3) nach $$\rho = \langle \boldsymbol{X}_2 - \boldsymbol{X}_1, \boldsymbol{e}_{12} \rangle \quad \text{und} \quad \dot{\rho} = \langle \dot{\boldsymbol{X}}_2 - \dot{\boldsymbol{X}}_1, \boldsymbol{e}_{12} \rangle \quad \text{mit} \quad \boldsymbol{e}_{12} = \frac{\boldsymbol{X}_2 - \boldsymbol{X}_1}{\lVert \boldsymbol{X}_2 - \boldsymbol{X}_1 \rVert}.$$	
Eingangsargumente		
`pos1`	$(n \times 3)$	Zeitreihe der Positionskomponenten des ersten Satelliten (GRACE A)
`pos2`	$(n \times 3)$	Zeitreihe der Positionskomponenten des zweiten Satelliten (GRACE B)
`vel1`	$(n \times 3)$	Zeitreihe der Geschwindigkeitskomponenten des ersten Satelliten (GRACE A)
`vel2`	$(n \times 3)$	Zeitreihe der Geschwindigkeitskomponenten des zweiten Satelliten (GRACE B)
Ausgabeargumente		
`rho`	$(n \times 1)$	Zeitreihe der Relativentfernung ρ
`rhop`	$(n \times 1)$	Zeitreihe der Relativgeschwindigkeit $\dot{\rho}$
`e12`	$(n \times 3)$	Zeitreihe des LOS-Einheitsvektors $\boldsymbol{e}_{12}$

Funktion A.8: `gradpshssp`

Aufgabe	Bestimmung des Gravitationsgradienten bezüglich eines lokalen Horizontsystems ($\nabla_h V$) bzw. des erdfesten Systems ($\nabla_e V$) für ein gegebenes Gravitationsfeldmodell an einem einzelnen Ort im Außenraum ($r \geq R_\mathrm{E},\ \varphi,\ \lambda$)	
Eingangsargumente		
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`lam`, `phi`, `r`	je (1×1)	sphärische, erdfeste Koordinaten des Berechnungspunkts (Länge, Breite, Radius)
`lmax`	(1×1)	Maximalgrad der Reihenentwicklung
`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_E)
`drtype`	string	Bezugssystem des ausgegebenen Vektors ∇V: lokal `'lscs'`, erdfest `'xyz'`.
`jflag`	boolean	falls 1, wird ein ellipsoidisches, zonales WGS 84-Normalenfeld (bestehend aus den Potentialkoeffizienten der Grade $0, 2, \ldots, \min\{8, l_\mathrm{max}\}$) vom gegebenen Gravitationsfeld subtrahiert (hier nicht erwünscht, daher 0 zu wählen für `jflag`)
Ausgabeargumente		
`gradV`	(1×3)	Gravitationsgradient ∇V bezüglich des gewählten Koordinatensystems

Funktion A.9: `kepler2cart`

Aufgabe	Umrechnung Keplerscher Bahnelemente in einen Positions- und Geschwindigkeitsvektor bezüglich des ebenen Perifokalsystems und bezüglich des raumfesten Äquatorsystems.	
Eingangsargumente		
`a`	(1×1)	große Halbachse a
`e`	(1×1)	Exzentrizität e
`I`	(1×1)	Inklination i
`omega`	(1×1)	Argument des Perigäums ω
`Omega`	(1×1)	Rektaszension des aufsteigenden Knotens Ω
`M`	(1×1)	mittlere Anomalie M
`GM`	(1×1)	Geozentrische Gravitationskonstante GM
`outinfo`	string	Falls `'num'`, erfolgt Ausgabe in numerischer anstatt symbolischer Darstellung
Ausgabeargumente		
`r_i`	(3×1)	Positionsvektor $\boldsymbol{X}$ im geozentrisch gelagerten, raumfesten Bezugssystem
`r_i_p`	(3×1)	Geschwindigkeitsvektor $\dot{\boldsymbol{X}}$ in ebendiesem System
`r_f`	(3×1)	Positionsvektor im ebenen, perifokalen Bezugssystem ($z_f \equiv 0$)
`r_f_p`	(3×1)	Geschwindigkeitsvektor in ebendiesem System ($\dot{z}_f \equiv 0$)

Funktion A.10: `mjd2greg`

Aufgabe	Umrechnung eines modifizierten julianischen Datums (MJD) in das entsprechende gregorianische Datum.	
Eingangsargumente		
`MJD`	(1×1)	MJD in Tagen, definiert durch MJD = JD − 2400000.5 wobei JD (julianisches Datum) die Anzahl der Tage sind seit 1. Jan. 4713 v. Chr. (12:00 Uhr Mittag)
Ausgabeargumente		
`timevec`	(1×6)	Gregorianisches Datum als Zeitvektor: [Jahr Monat Tag Stunde Min. Sek.]

Funktion A.11: `orbint_field`

Aufgabe		Orbitintegration für ein gegebenes Gravitationsfeldmodell ausgehend von einem bekannten initialen Bahnvektor $\hat{\boldsymbol{s}}_0$ inklusive Propagation der zugehörigen Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ bzw. der Transitionsmatrix $\boldsymbol{\Phi}$ (vgl. Abschn. 3.2.5.1, S. 45 ff.).
Eingangsargumente		
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`tspan`	$(n \times 1)$	Vektor der Prädiktionszeitpunkte, z.B. $[0, 1, \dots, 60]$ Sek.
`X0`	(1×6)	Startvektor der Orbitintegration $\hat{\boldsymbol{s}}_0$ (Anfangsposition und -geschwindigkeit im quasi-inertialen Bezugssystem)
`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_E)
`Ri2e`	$(r \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$, Matrixstruktur s. Abschn. 4.1.1
`lmax`	(1×1)	Maximalgrad bei der Reihenentwicklung für $\nabla_h V$
`cov_sel`	string	Wahl der als Zeitreihe `mat` ausgegebenen Matrix (`'ric'` oder `'var'`)
`lmax2`	(1×1)	Maximalgrad bei der Berechnung des Marussi-Tensors $\boldsymbol{M}_h$ (empfohlen < 12)
`tens_sel`	string	Falls `'PetVer'`, Berechnung von $\boldsymbol{M}_e$ durch `tenspshs`, ansonsten durch `tenssp`
`P0`	(6×6)	Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ des Startvektors (nur benötigt falls `cov_sel` = `'ric'`)
`Q`	(6×6)	Spektraldichtematrix des Systemrauschens (nur benötigt falls `cov_sel` = `'ric'`)
Ausgabeargumente		
`X`	$(n \times 6)$	Zeitreihe der integrierten Bahnvektoren $\bar{\boldsymbol{s}}$
`mat`	$(n \times 36)$	Zeitreihe der Kovarianzmatrizen $\overline{\boldsymbol{P}}$ (falls `cov_sel` = `'ric'`) bzw. der Transitionsmatrizen $\boldsymbol{\Phi}$ (falls `cov_sel` = `'var'`)

Funktion A.12: `orbint_field2`

Aufgabe		Orbitintegration für ein gegebenes Gravitationsfeldmodell ausgehend von einem bekannten initialen Bahnvektor $\hat{\boldsymbol{s}}_0$. Propagation der Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ bzw. der Transitionsmatrix $\boldsymbol{\Phi}$ jedoch wie im J_2-gestörten Fall (`orbint_j2`).
Eingangsargumente		
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`tspan`	$(n \times 1)$	Vektor der Prädiktionszeitpunkte, z.B. $[0, 1, \dots, 60]$ Sek.
`X0`	(1×6)	Startvektor der Orbitintegration $\hat{\boldsymbol{s}}_0$ (Anfangsposition und -geschwindigkeit im quasi-inertialen Bezugssystem)
`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_E, J_2)
`Ri2e`	$(r \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$, Matrixstruktur s. Abschn. 4.1.1
`lmax`	(1×1)	Maximalgrad bei der Reihenentwicklung für $\nabla_h V$
`cov_sel`	string	Wahl der als Zeitreihe `mat` ausgegebenen Matrix (`'ric'` oder `'var'`)
`P0`	(6×6)	Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ des Startvektors (nur benötigt falls `cov_sel` = `'ric'`)
`Q`	(6×6)	Spektraldichtematrix des Systemrauschens (nur benötigt falls `cov_sel` = `'ric'`)
Ausgabeargumente		
`X`	$(n \times 6)$	Zeitreihe der integrierten Bahnvektoren $\bar{\boldsymbol{s}}$
`mat`	$(n \times 36)$	Zeitreihe der Kovarianzmatrizen $\overline{\boldsymbol{P}}$ (falls `cov_sel` = `'ric'`) bzw. der Transitionsmatrizen $\boldsymbol{\Phi}$ (falls `cov_sel` = `'var'`)

Funktion A.13: `orbint_j2`

Aufgabe		Orbitintegration für ein Gravitationsfeldmodell bis einschließlich Grad 2 (Zentralfeld & J_2-Anteil) ausgehend von einem bekannten initialen Bahnvektor $\hat{\boldsymbol{s}}_0$ inklusive Propagation der Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ bzw. der Transitionsmatrix $\boldsymbol{\Phi}$ (vgl. Abschn. 3.2.5.2, S. 51).
Eingangsargumente		
`tspan`	$(n \times 1)$	Vektor der Prädiktionszeitpunkte, , z.B. $[0, 1, \dots, 60]$ Sek.
`X0`	(1×6)	Startvektor der Orbitintegration $\hat{\boldsymbol{s}}_0$ (Anfangsposition und -geschwindigkeit im quasi-inertialen Bezugssystem)

Fortsetzung `orbint_j2`

`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_{E}, J_2)
`cov_sel`	string	Wahl der als Zeitreihe `mat` ausgegebenen Matrix (`'ric'` oder `'var'`)
`P0`	(6×6)	Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ des Startvektors (nur benötigt falls `cov_sel` = `'ric'`)
`Q`	(6×6)	Spektraldichtematrix des Systemrauschens (nur benötigt falls `cov_sel` = `'ric'`)
Ausgabeargumente		
`X`	$(n \times 6)$	Zeitreihe der integrierten Bahnvektoren $\bar{\boldsymbol{s}}$
`mat`	$(n \times 36)$	Zeitreihe der Kovarianzmatrizen $\bar{\boldsymbol{P}}$ (falls `cov_sel` = `'ric'`) bzw. der Transitionsmatrizen $\boldsymbol{\Phi}$ (falls `cov_sel` = `'var'`)

Funktion A.14: `orbint_kep`

Aufgabe	Bestimmung eines Keplerorbits, der am Bahnvektor $\hat{\boldsymbol{s}}_0$ beginnt, durch numerische Integration über alle Zeitpunkte `tspan`. Propagation der Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ bzw. der Transitionsmatrix $\boldsymbol{\Phi}$ ebenfalls durch numerische Integration (vgl. Abschn. 3.2.5.3, S. 54).	
Eingangsargumente		
`tspan`	$(n \times 1)$	Vektor der Prädiktionszeitpunkte, z.B. $[0, 1, \ldots, 60]$ Sek.
`X0`	(1×6)	Startvektor der Orbitintegration $\hat{\boldsymbol{s}}_0$ (Anfangsposition und -geschwindigkeit im quasi-inertialen Bezugssystem)
`type`	string/struct	zu verwendende Konstanten (GM)
`cov_sel`	string	Wahl der als Zeitreihe `mat` ausgegebenen Matrix (`'ric'` oder `'var'`)
`P0`	(6×6)	Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ des Startvektors (nur benötigt falls `cov_sel` = `'ric'`)
`Q`	(6×6)	Spektraldichtematrix des Systemrauschens (nur benötigt falls `cov_sel` = `'ric'`)
Ausgabeargumente		
`X`	$(n \times 6)$	Zeitreihe der integrierten Bahnvektoren $\bar{\boldsymbol{s}}$
`mat`	$(n \times 36)$	Zeitreihe der Kovarianzmatrizen $\bar{\boldsymbol{P}}$ (falls `cov_sel` = `'ric'`) bzw. der Transitionsmatrizen $\boldsymbol{\Phi}$ (falls `cov_sel` = `'var'`)

Funktion A.15: `orbint_kep2`

Aufgabe	Bestimmung eines Keplerorbits, der am Bahnvektor $\hat{\boldsymbol{s}}_0$ beginnt, durch Auswertung analytischer Formeln für alle Zeitpunkte `tspan`. Falls `cov_sel = 'var'` gewählt ist, werden die Zeitreihen $\bar{\boldsymbol{s}}$ und $\boldsymbol{\Phi}$ mittels der Lagrange/Gibbs-Lösung vollkommen analytisch berechnet (mittels `fgkepler`). Ansonsten $\bar{\boldsymbol{s}}$ mittels Ephemeridenrechnung und $\widehat{\boldsymbol{P}}_0$ durch numerische Integration (vgl. Abschn. 3.2.5.4, S. 55).	
Eingangsargumente		
`tspan`	$(n \times 1)$	Vektor der Prädiktionszeitpunkte, z.B. $[0, 1, \ldots, 60]$ Sek.
`X0`	(1×6)	Startvektor der Orbitintegration $\hat{\boldsymbol{s}}_0$ (Anfangsposition und -geschwindigkeit im quasi-inertialen Bezugssystem)
`type`	string/struct	zu verwendende Konstanten (GM)
`cov_sel`	string	Wahl der als Zeitreihe `mat` ausgegebenen Matrix (`'ric'` oder `'var'`)
`P0`	(6×6)	Kovarianzmatrix $\widehat{\boldsymbol{P}}_0$ des Startvektors (nur benötigt falls `cov_sel` = `'ric'`)
`Q`	(6×6)	Spektraldichtematrix des Systemrauschens (nur benötigt falls `cov_sel` = `'ric'`)
Ausgabeargumente		
`X`	$(n \times 6)$	Zeitreihe der integrierten Bahnvektoren $\bar{\boldsymbol{s}}$
`mat`	$(n \times 36)$	Zeitreihe der Kovarianzmatrizen $\bar{\boldsymbol{P}}$ (falls `cov_sel` = `'ric'`) bzw. der Transitionsmatrizen $\boldsymbol{\Phi}$ (falls `cov_sel` = `'var'`)

Funktion A.16: `plmsp`

Aufgabe	Rekursive Berechnung aller vollständig normierten Legendre-Funktionen $\bar{P}_{l,m}(\sin(\varphi))$ bis zum Grad $l_{\max}$ für eine einzelne gegebene Ko-Breite θ. Zudem Bestimmung der zugehörigen ersten und zweiten partiellen Ableitungen $\left.\frac{\partial \bar{P}_{l,m}(\sin(\varphi))}{\partial \varphi}\right\rvert_{\theta}$ und $\left.\frac{\partial^2 \bar{P}_{l,m}(\sin(\varphi))}{\partial \varphi^2}\right\rvert_{\theta}$	
Eingangsargumente		
`lmax`	(1×1)	maximaler sphärisch-harmonischer Grad $l_{\max}$
`th`	(1×1)	Ko-Breite $\theta = 90° - \varphi$
Ausgabeargumente		
`p`	$(l_{\max}+1 \times 2\,l_{\max}+1)$	Matrix mit vollst. normierten Legendre-Funktionen $\bar{P}_{l,m}(\sin(\varphi))$ im SC-Format
`dp`	$(l_{\max}+1 \times 2\,l_{\max}+1)$	Matrix mit ersten Ableitungen der Legendre-Funktionen nach φ im SC-Format
`ddp`	$(l_{\max}+1 \times 2\,l_{\max}+1)$	Matrix mit zweiten Ableitungen der Legendre-Funktionen nach φ im SC-Format

Funktion A.17: `rot_statevec`

Aufgabe	Transformation einer Zeitreihe von Zustandsvektoren (Bahnvektoren) zwischen dem quasi-inertialen und dem erdfesten Bezugssystem für gegebene Rotationsmatrizen.	
Eingangsargumente		
`info`	string	Wahl der Transformationsrichtung: `'e2i'` (erdfest $\longrightarrow$ quasi-inertial) oder `'i2e'` (quasi-inertial $\longrightarrow$ erdfest)
`R`	$(n \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$ bzw. $\boldsymbol{R}_i^e$, Matrixstruktur s. Abschn. 4.1.1
`pos`	$(n \times 3)$	Zeitreihe der Positionskomponenten x, y, z bzw. X, Y, Z
`vel`	$(n \times 3)$	Zeitreihe der Geschwindigkeitskomponenten $\dot{x}, \dot{y}, \dot{z}$ bzw. $\dot{X}, \dot{Y}, \dot{Z}$
`omega`	$(n \times 3)$	Zeitreihe der Winkelgeschwindigkeiten der Erde $\boldsymbol{\omega} = (\omega_1,\ \omega_2,\ \omega_3)^\top$
Ausgabeargumente		
`pos_r`	$(n \times 3)$	Zeitreihe der rotierten Positionskomponenten X, Y, Z bzw. x, y, z
`vel_r`	$(n \times 3)$	Zeitreihe der rotierten Geschwindigkeitskomponenten $\dot{X}, \dot{Y}, \dot{Z}$ bzw. $\dot{x}, \dot{y}, \dot{z}$

Funktion A.18: `rotation_ts`

Aufgabe	Berechnung einer Zeitreihe von Rotationsmatrizen um eine wählbare Achse des als rechtshändig angenommenen Koordinatensystems für eine gegebene Zeitreihe von Rotationswinkeln.	
Eingangsargumente		
`alpha`	$(n \times 1)$	Zeitreihe der Rotationswinkel
`rot_axis`	(1×1)	Rotationsachse (1, 2 oder 3)
Ausgabeargumente		
`R`	$(n \times 9)$	Zeitreihe der Rotationsmatrizen

Funktion A.19: `shootorb`

Aufgabe	Korrektur der Anfangsbedingungen eines gegebenen Satellitenorbits durch Schießverfahren (mithilfe `fsolve`) oder Schätzung einer ausgleichenden Bahn (mithilfe `lsqnonlin`). Anschließend Orbitintegration mittels `orbint_field` ausgehend vom korrigierten Zustandsvektor (s.a. Abschnitt 2.3.4, S. 20 ff. und Glättungsmethoden 2 und 3 ab S. 125)	
Eingangsargumente		
`pos`	$(n \times 5)$	Zeitreihe der quasi-inertialen Positionskomponenten. Spalten: [MJD t X Y Z]
`vel`	$(n \times 5)$	Zeitreihe der quasi-inertialen Geschwindigkeitskomponenten: [MJD t $\dot{X}$ $\dot{Y}$ $\dot{Z}$]

Fortsetzung `shootorb`

`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`time`	$(m \times 1)$	Vektor der Prädiktionszeitpunkte, z.B. $[0, 1, \ldots, 60]$ Sek.
`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_E)
`Ri2e`	$(r \times 9)$	Zeitreihe der Rotationsmatrizen $\boldsymbol{R}_e^i$, Matrixstruktur s. Abschn. 4.1.1
`lmax`	(1×1)	Maximalgrad bei der Reihenentwicklung für $\nabla_h V$
`tolx`	(1×1)	Abbruchkriterium bzgl. der Änderung der Korrektur (s.a. Beschr. S. 125)
`tolfun`	(1×1)	Abbruchkriterium bzgl. der Änderung der Zielfunktion (s.a. Beschr. S. 125)
`info`	string	definiert Optimierungmethode (`fsolve`/`lsqnonlin`) und legt fest, welche Komponenten des Anfangsvektors korrigiert werden (Details s. Quellcode)
Ausgabeargumente		
`intpos`	$(m \times 5)$	Zeitreihe der integrierten quasi-inertialen Positionskomponenten. Spalten wie `pos`
`intvel`	$(m \times 5)$	Zeitreihe der integrierten quasi-inertialen Geschwindigkeitskomp. Spalten wie `vel`
`ini_state`	(1×6)	korrigierter Startvektor, der zur Orbitintegration verwendet wurde
`exitflag`	(1×1)	Codenummer zur Identifikation der Ursache eines eventuellen Abbruchs (weitergegeben vom Löser `fsolve`/`lsqnonlin`, Näheres s. Quellcode)
`output`	struct	Informationen über die erfolgte Optimierung (erhalten vom Lösungsalgorithmus)

Funktion A.20: `slerp_mat`

Aufgabe	Berechnung einer DCM an der Stelle `t` zwischen zwei gegebenen Rotationsmatrizen durch sphärisch-lineare Interpolation (s.a. Anhang B.1, S. 145 ff.)	
Eingangsargumente		
`mata`	(3×3)	DCM am ‚linken Rand' (Standpunkt)
`matb`	(3×3)	DCM am ‚rechten Rand' (Endpunkt)
`t`	(3×3)	Interpolationspunkt `t` $\in [0, 1]$
`eps`	(3×3)	Schwellwert unterhalb dessen die Interpolation linear erfolgt (Details hierzu im Quellcode bzw. in Anhang B.1)
Ausgabeargumente		
`mat_i`	(3×3)	interpolierte DCM

Funktion A.21: `tenspshs`

Aufgabe	Bestimmung des Marussi-Tensors $\boldsymbol{M}_\mathrm{L}$ bzw. $\boldsymbol{M}_\mathrm{E}$ (zweite Ableitungen des Gravitationspotentials nach kartesischen Koordinaten x, y, z im lokalen Horizontsystem bzw. im erdfesten System) für ein gegebenes Gravitationsfeldmodell, ausgewertet an mehreren Berechnungspunkten im Außenraum (Zeitreihen für r, φ, λ). Die Berechnungen basieren auf dem nicht-singulären Verfahren nach PETROVSKAYA und VERSHKOV. Der zurückgegebene Tensor umfasst nur die Grade von 2 bis l_max, d.h. Grad 0 (und 1) bleibt unberücksichtigt, sodass der Tensor des Zentralfelds gegebenenfalls außerhalb gemäß Gl. (3.102), S. 55 zu berechnen und aufzuaddieren ist.	
Eingangsargumente		
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`lam`, `phi`, `r`	je $(t \times 1)$	sphärische, erdfeste Koordinaten der Berechnungspunkte (Länge, Breite, Radius)
`lmax`	(1×1)	Maximalgrad der Reihenentwicklung
`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_E)
`drtype`	string	Bezugssystem des ausgegebenen Tensors $\boldsymbol{M}$: lokal `'lscs'`, erdfest `'xyz'`.
`info`	(1×1)	Zugriffsverweis auf Fortschrittsanzeige („wait bar"). Mit `[]` ist diese deaktiviert.
`jflag`	boolean	falls 1, wird ein ellipsoidisches, zonales WGS 84-Normalenfeld (bestehend aus den Potentialkoeffizienten der Grade $0, 2, \ldots, \min\{8, l_\mathrm{max}\}$) vom gegebenen Gravitationsfeld subtrahiert (hier nicht erwünscht, daher 0 zu wählen für `jflag`)
Ausgabeargumente		
`tensV`	$(t \times 9)$	Marussi-Tensor(en) $\boldsymbol{M} = \nabla^\top \otimes \nabla\,[V(r, \varphi, \lambda)]$ bezüglich des gewählten Koordinatensystems (in $\mathrm{E} \mathrel{\hat{=}} 10^{-9}\,\mathrm{s}^{-2}$). Spalten: $[V_{xx}\ V_{xy}\ V_{xz}\ V_{yx}\ V_{yy}\ V_{yz}\ V_{zx}\ V_{zy}\ V_{zz}]$

Funktion A.22: `tenssp`

Aufgabe	Bestimmung des Marussi-Tensors $\boldsymbol{M}_\mathrm{L}$ bzw. $\boldsymbol{M}_\mathrm{E}$ (zweite Ableitungen des Gravitationspotentials nach kartesischen Koordinaten x, y, z im lokalen Horizontsystem bzw. im erdfesten System) für ein gegebenes Gravitationsfeldmodell, ausgewertet an einem einzelnen Berechnungspunkt im Außenraum ($r \geq R_\mathrm{E},\ \varphi,\ \lambda$). Dabei werden die Beziehungen (3.67) bis (3.69) sowie Gl. (3.72) bis (3.79) genutzt (s. S. 49 ff.).	
Eingangsargumente		
`field`	$(u \times v)$	vollständig normierte Potentialkoeffizienten des Gravitationsfeldmodells im CS- oder SC-Format
`lam`, `phi`, `r`	je (1×1)	sphärische, erdfeste Koordinaten des Berechnungspunkts (Länge, Breite, Radius)
`lmax`	(1×1)	Maximalgrad der Reihenentwicklung
`type`	string/struct	Konstanten des Gravitationsfeldmodells (GM, a_E)
`drtype`	string	Bezugssystem des ausgegebenen Tensors $\boldsymbol{M}$: lokal `'lscs'`, erdfest `'xyz'`.
`jflag`	boolean	falls 1, wird ein ellipsoidisches, zonales WGS 84-Normalenfeld (bestehend aus den Potentialkoeffizienten der Grade $0, 2, \ldots, \max\{8, l_\mathrm{max}\}$) vom gegebenen Gravitationsfeld subtrahiert (hier nicht erwünscht, daher 0 zu wählen für `jflag`)
Ausgabeargumente		
`tensV`	(3×3)	Marussi-Tensor $\boldsymbol{M} = \nabla^\top \otimes \nabla\,[V(r, \varphi, \lambda)]$ bezüglich des gewählten Koordinatensystems (in s^{-2}).

Anhang B

B Ergänzende Anmerkungen

B.1 Sphärisch-lineare Interpolation

Bei der Orbitintegration mit `orbint_field` und `orbint_field2` müssen die Rotationsmatrizen, die nur in bestimmten zeitlichen Abständen vorliegen, am aktuellen, vom DGL-Löser vorgegebenen Integrationszeitpunkt gefunden werden. Ein schneller aber dennoch zweckmäßiger Lösungsweg scheint zu sein, die zwei betreffenden Rotationsmatrizen in Quaternionen zu überführen, zwischen diesen sphärisch-linear zu interpolieren und das interpolierte Quaternion wieder in eine Rotationsmatrix umzurechnen. Umgesetzt wurde dies in der Funktion `slerp_mat`. Im Folgenden soll nun der Lösungsweg inklusive der verwendeten Formeln dargelegt werden.

Vorab sei angemerkt, dass es sich bei einem Quaternion bekanntlich um ein Quadrupel reeller Zahlen

$$\boldsymbol{q} \overset{\text{def}}{=} (q_0,\ q_1,\ q_2,\ q_3)^\top = \begin{pmatrix} q_0 \\ \widetilde{\boldsymbol{q}} \end{pmatrix} \tag{B.1}$$

handelt, welches man in den skalaren Anteil q_0 und den dreidimensionalen Vektor $\widetilde{\boldsymbol{q}}$ einteilen kann. Die Theorie der Quaternionen wurde von Sir William Rowan Hamilton (1805–1865) entwickelt und kann im Prinzip als Erweiterung der komplexen Zahlen verstanden werden. Üblicherweise bezieht man Quaternionen stets auf ihre Einheitslänge ($\langle \boldsymbol{q}, \boldsymbol{q} \rangle = 1$), d.h. ihr Freiheitsgrad ist (wie bei einer Rotationsmatrix) auf drei beschränkt. Detaillierte Informationen zur Theorie der Quaternionen findet man etwa in Hanson (2006).

Gegeben sind nun also zwei (3×3)-Rotationsmatrizen[1] für ein rechtshändiges Koordinatensystem, $\boldsymbol{R}_a = \boldsymbol{R}(t_a)$ und $\boldsymbol{R}_b = \boldsymbol{R}(t_b)$. Gesucht ist die interpolierte Rotationsmatrix $\boldsymbol{R}(t_i)$, $t_a < t_i < t_b$.

Schritt 1 – Berechnung der Quaternionen $\boldsymbol{q}_a$ und $\boldsymbol{q}_b$

Die zwei gegebenen Rotationsmatrizen müssen zunächst in Quaternionendarstellung überführt werden, wofür es unterschiedliche Techniken gibt. Hier kommt die u.a. in Shuster (1993, S. 463) und Dever (1998, S. 153) beschriebene Methode zum Einsatz, da sie gut implementierbar ist (so geschehen in der Unterfunktion `mat2q` der Funktion `slerp_mat`). Demnach sind zuerst die vier Größen

$$\alpha_1 = 1 + \operatorname{tr}(\boldsymbol{R}) \tag{B.2}$$

$$\alpha_2 = 1 + R_{11} - R_{22} - R_{33} \tag{B.3}$$

$$\alpha_3 = 1 - R_{11} + R_{22} - R_{33} \tag{B.4}$$

$$\alpha_4 = 1 - R_{11} - R_{22} + R_{33} \tag{B.5}$$

aus den Einträgen der gegebenen Rotationsmatrix $\boldsymbol{R}$ zu berechnen.

1 Auch: „direction cosine matrix“, DCM.

Den Autoren zufolge können nun die Einträge des Quaternions nach vier analytisch äquivalenten Rechenwegen aus der DCM bestimmt werden. Man wähle dabei denjenigen Weg, bei dem α_j am größten ist ($j \in \{1, \ldots, 4\}$):

Falls $\alpha_1 = \max\{\alpha_1, \ldots, \alpha_4\}$

$$q_0 = \frac{1}{2}\sqrt{\alpha_1} \tag{B.6}$$

$$q_1 = \frac{1}{4q_0}(R_{23} - R_{32}) \tag{B.7}$$

$$q_2 = \frac{1}{4q_0}(R_{31} - R_{13}) \tag{B.8}$$

$$q_3 = \frac{1}{4q_0}(R_{12} - R_{21}) \tag{B.9}$$

Falls $\alpha_2 = \max\{\alpha_1, \ldots, \alpha_4\}$

$$q_1 = \frac{1}{2}\sqrt{\alpha_2} \tag{B.10}$$

$$q_0 = \frac{1}{4q_1}(R_{23} - R_{32}) \tag{B.11}$$

$$q_2 = \frac{1}{4q_1}(R_{12} + R_{21}) \tag{B.12}$$

$$q_3 = \frac{1}{4q_1}(R_{13} + R_{31}) \tag{B.13}$$

Falls $\alpha_3 = \max\{\alpha_1, \ldots, \alpha_4\}$

$$q_2 = \frac{1}{2}\sqrt{\alpha_3} \tag{B.14}$$

$$q_0 = \frac{1}{4q_2}(R_{31} - R_{13}) \tag{B.15}$$

$$q_1 = \frac{1}{4q_2}(R_{12} + R_{21}) \tag{B.16}$$

$$q_3 = \frac{1}{4q_2}(R_{23} + R_{32}) \tag{B.17}$$

Falls $\alpha_4 = \max\{\alpha_1, \ldots, \alpha_4\}$

$$q_3 = \frac{1}{2}\sqrt{\alpha_4} \tag{B.18}$$

$$q_0 = \frac{1}{4q_3}(R_{12} - R_{21}) \tag{B.19}$$

$$q_1 = \frac{1}{4q_3}(R_{13} + R_{31}) \tag{B.20}$$

$$q_2 = \frac{1}{4q_3}(R_{23} + R_{32}) \tag{B.21}$$

Unter den vier Möglichkeiten ist gerade die so gewählte die numerisch vorteilhafteste, denn die Wurzel aus α_j (jeweils erste Gleichung) ist dann am größten, sodass folglich auch der Nenner in den übrigen drei Gleichungen maximal, d.h. am ehesten von Null verschieden ist.

Da Rundungsfehler zur Nicht-Orthogonalität führen können, normalisiert man das berechnete Quaternion noch zu

$$\boldsymbol{q}_\mathrm{n} = \frac{\boldsymbol{q}}{\sqrt{\langle \boldsymbol{q}, \boldsymbol{q} \rangle}}. \tag{B.22}$$

Schritt 2 – Berechnung von $\boldsymbol{q}_i$ über sphärisch-lineare Interpolation

Das sphärisch-linear interpolierte Quaternion ergibt sich am relativen Zeitpunkt $u \in [0, 1]$ zwischen dem ersten Quaternion $\boldsymbol{q}(t_a)$ und dem zweiten Quaternion $\boldsymbol{q}(t_b)$ nach der Beziehung

$$\boldsymbol{q}(t_i) = \boldsymbol{q}(t_a)\frac{\sin(1-u)\theta}{\sin\theta} + \boldsymbol{q}(t_b)\frac{\sin u\theta}{\sin\theta}, \tag{B.23}$$

worin wir den Winkel zwischen den Quaternionen $\theta = \sphericalangle(\boldsymbol{q}_a, \boldsymbol{q}_b)$ aus dem Skalarprodukt

$$\cos\theta = \langle \boldsymbol{q}(t_a), \boldsymbol{q}(t_b) \rangle. \tag{B.24}$$

erhalten. Im Gegensatz zum zweidimensionalen Fall findet die sphärisch-lineare Interpolation[1], die wohl erstmals von Shoemake (1985, Abschn. 3.3) veröffentlicht wurde, nicht auf einer

1 Engl.: „spherical linear interpolation“, SLERP

Geraden, sondern auf demjenigen Großkreis der Einheitskugel statt, der die kürzeste Verbindung zwischen den Interpolationspunkten aufweist, d.h. die Interpolationskurve ist gerade die Geodätische zwischen den zwei gegebenen Quaternionen. Als weitere vorteilhafte Eigenschaft ist zu nennen, dass der SLERP-Algorithmus einer Rotation mit konstanter Winkelgeschwindigkeit um eine feste Raumachse entspricht.

Jedoch birgt die Interpolation nach Vorschrift (B.23) auch Risiken. So können sehr kleine Winkel θ wegen dem Ausdruck $\sin\theta$ im Nenner numerische Probleme verursachen (Division durch Null). Daher lässt sich in der Funktion `slerp_mat` ein Schwellwert `eps` für den Ausdruck $1-\cos\theta$ vorgeben, unterhalb dessen $\boldsymbol{q}(t_i)$ anstatt nach Gl. (B.23) unmittelbar durch die lineare Interpolation

$$\boldsymbol{q}(t_i) = \boldsymbol{q}(t_a) + u\big(\boldsymbol{q}(t_b) - \boldsymbol{q}(t_a)\big) \tag{B.25}$$

bestimmt wird. Für `eps` wurde in den Funktionen `orbint_field(2)` der Wert 10^{-10} gewählt. Dies entspricht etwa $2\overset{''}{.}9$ (Bogensekunden) für den Winkel θ.

Schritt 3 – Berechnung der zu $\boldsymbol{q}_i$ gehörenden DCM

In der Unterfunktion `q2mat` stellt man durch erneutes Anwenden der Gleichung (B.22) sicher, dass $\boldsymbol{q}_i$ ein Einheitsquaternion ist. Schließlich erhält man die zugehörige DCM mithilfe folgender Beziehung, die bereits zur Herleitung der Gleichungen B.6 bis B.21 genutzt wurde (vgl. Shuster, 1993, S. 462, Gl. (158)):

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix} \tag{B.26}$$

B.2 Transformation des Zustandsvektors zwischen erd- und raumfestem Bezugssystem

Der Zustandsvektor, der in Abschnitt 3.2.1 in der Systemgleichung eingeführt wurde, besteht aus drei Positions- und drei Geschwindigkeitskomponenten des Satelliten im geozentrisch gelagerten, raumfesten Bezugssystem:

$$\boldsymbol{s}(t) = (\boldsymbol{X},\ \dot{\boldsymbol{X}})^\top = (X,\ Y,\ Z,\ \dot{X},\ \dot{Y},\ \dot{Z})^\top$$

Bisweilen ist es in den Matlab-Funktionen erforderlich, den Zustandsvektor (oder nur dessen Positionskomponenten) in das terrestrische, geozentrisch gelagerte Bezugssystem zu überführen bzw. umgekehrt vom erdfesten in das raumfeste System zu transformieren. In der Toolbox dient dazu die Funktion `rot_statevec`. Nachfolgend werden die dabei genutzten Formeln erläutert.

Den erdfesten, dreidimensionalen Positionsvektor $\boldsymbol{x}$ erhält man durch einfache Multiplikation einer geeigneten (3×3)-Rotationsmatrix $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}(\alpha(t), \beta(t), \gamma(t))$ mit dem raumfesten Positionsvektor gemäß

$$\boldsymbol{x} = \boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}\boldsymbol{X}. \tag{B.27}$$

Es sei hier fortan $\boldsymbol{R}_{\mathrm{E}}^{\mathrm{I}}$ abgekürzt durch $\boldsymbol{R}$. Somit gilt wegen der Orthogonalität der Rotationsmatrix für die umgekehrte Transformation

$$\boldsymbol{X} = \boldsymbol{R}^{-1}\boldsymbol{x} = \boldsymbol{R}^{\top}\boldsymbol{x}. \tag{B.28}$$

Um die Transformationsbeziehung für die Geschwindigkeitsvektoren zu erhalten, ist die Zeitableitung der letzten, im quasi-inertialen System formulierten Gleichung zu bilden:

$$\dot{\boldsymbol{X}} = \dot{\boldsymbol{R}}^{\top}\boldsymbol{x} + \boldsymbol{R}^{\top}\dot{\boldsymbol{x}} \quad \xRightarrow{\cdot \boldsymbol{R}} \quad \boldsymbol{R}\dot{\boldsymbol{X}} = \boldsymbol{R}\dot{\boldsymbol{R}}^{\top}\boldsymbol{x} + \boldsymbol{R}\boldsymbol{R}^{\top}\dot{\boldsymbol{x}} = \boldsymbol{\Omega}\boldsymbol{x} + \dot{\boldsymbol{x}}$$

Hierin ist $\boldsymbol{\Omega} = \boldsymbol{R}\dot{\boldsymbol{R}}^{\top}$ die sog. Cartan-Matrix[1], welche die zeitliche Änderung der Rotation, d.h. die Drehraten, beschreibt. Mithilfe des Drehratenvektors $\boldsymbol{\omega}$ kann das Produkt $\boldsymbol{\Omega}\boldsymbol{x}$ auch geschrieben werden kann als (vgl. Talman, 2007, Gl. 6.73; Sneeuw, 2006, Abschn. 4.1, S. 37 ff.):

$$\boldsymbol{\Omega}\boldsymbol{x} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{\omega} \times \boldsymbol{x}.$$

Zusammen mit der vorigen Gleichung folgt daraus

$$\dot{\boldsymbol{x}} = \boldsymbol{R}\dot{\boldsymbol{X}} - \boldsymbol{\omega} \times \boldsymbol{x} \tag{B.29}$$

bzw.

$$\dot{\boldsymbol{X}} = \boldsymbol{R}^{\top}\left(\dot{\boldsymbol{x}} + \boldsymbol{\omega} \times \boldsymbol{x}\right). \tag{B.30}$$

In der Funktion `rot_statevec` nutzt man Vorschrift (B.27) bis (B.30), um Zeitreihen von Zustandsvektoren zwischen dem raumfesten und dem rotierenden Bezugssystem zu transformieren.

Durch zweimaliges Differenzieren der Gl. (B.28) nach der Zeit lässt sich im Übrigen analog die bekannte Beziehung

$$\boldsymbol{R}\ddot{\boldsymbol{X}} = \ddot{\boldsymbol{x}} + 2\boldsymbol{\omega} \times \dot{\boldsymbol{x}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{x}) + \dot{\boldsymbol{\omega}} \times \boldsymbol{x}$$

für die Beschleunigungen mit dem Coriolis-, Zentrifugal- und Euler-Term herleiten (z.B. Talman, 2007, Gl. 6.74; Seeber, 1989, S. 107; Sneeuw, 2006, Gl. 4.5b, S. 40).

B.3 Gradientenbildung des J_2-Störpotentials

In Abschnitt 3.2.5.2 (Seite 51 ff.) wird der Zentralterm in der Kraftfunktion um den J_2-Anteil ergänzt gemäß

$$\boldsymbol{f}_{\mathrm{K}}(\boldsymbol{X}(t), \boldsymbol{p}) = \nabla V_2(\boldsymbol{X}(t), \boldsymbol{p}) = \nabla \frac{GM}{r} + \nabla R_{2,0}(\boldsymbol{X}(t), \boldsymbol{p}) = -\frac{GM}{r^3}\boldsymbol{X} + \nabla R_{2,0}(\boldsymbol{X}(t), \boldsymbol{p})$$

1 Benannt nach dem französischen Mathematiker Élie Joseph Cartan (1869-1951).

Mit Gleichung (3.82) für das Störpotential $R_{2,0}$ ergibt sich der gesuchte Gradient aus

$$\begin{aligned}\nabla R_{2,0}(\boldsymbol{X}(t),\boldsymbol{p}) &= \nabla\left(\frac{1}{2}\cdot\frac{GM}{R_\mathrm{E}}\cdot\frac{R_\mathrm{E}^3}{r^5}\cdot C_{2,0}\cdot(2Z^2-X^2-Y^2)\right)\\ &= \frac{1}{2}\cdot\frac{GM}{R_\mathrm{E}}\cdot R_\mathrm{E}^3\cdot C_{2,0}\cdot\begin{pmatrix}\frac{\partial}{\partial X}\\ \frac{\partial}{\partial Y}\\ \frac{\partial}{\partial Z}\end{pmatrix}\left(\frac{1}{r^5}\cdot(2Z^2-X^2-Y^2)\right).\end{aligned}\tag{B.31}$$

Für die erste Komponente erhält man

$$\begin{aligned}\frac{\partial}{\partial X}\left[\frac{1}{r^5}\cdot(2Z^2-X^2-Y^2)\right] &= \frac{\partial}{\partial r}\left[\frac{1}{r^5}\right]\cdot\frac{\partial r}{\partial X}\cdot(2Z^2-X^2-Y^2)+\frac{1}{r^5}\cdot\frac{\partial}{\partial X}\left[2Z^2-X^2-Y^2\right]\\ &= -\frac{5}{r^6}\cdot\frac{X}{r}\cdot(2Z^2-X^2-Y^2)+\frac{1}{r^5}\cdot(-2X)\\ &= \frac{X}{r^5}\left(-\frac{5}{r^2}\cdot(2Z^2-X^2-Y^2-Z^2+Z^2)-2\right)\\ &= \frac{X}{r^5}\left(-\frac{5}{r^2}\cdot(3Z^2-r^2)-2\right) = \frac{X}{r^5}\left(-5\cdot\frac{3Z^2}{r^2}+5-2\right)\\ &= \frac{3X}{r^5}\left(-5\cdot\frac{Z^2}{r^2}+1\right),\end{aligned}\tag{B.32}$$

und für die zweite Komponente entsprechend

$$\frac{\partial}{\partial Y}\left[\frac{1}{r^5}\cdot(2Z^2-X^2-Y^2)\right] = \frac{3Y}{r^5}\left(-5\cdot\frac{Z^2}{r^2}+1\right).\tag{B.33}$$

Die Ableitung nach der dritten Komponente Z berechnet man wie folgt:

$$\begin{aligned}\frac{\partial}{\partial Z}\left[\frac{1}{r^5}\cdot(2Z^2-X^2-Y^2)\right] &= \frac{\partial}{\partial r}\left[\frac{1}{r^5}\right]\cdot\frac{\partial r}{\partial Z}\cdot(2Z^2-X^2-Y^2)+\frac{1}{r^5}\cdot\frac{\partial}{\partial Z}\left[2Z^2-X^2-Y^2\right]\\ &= -\frac{5}{r^6}\cdot\frac{Z}{r}\cdot(2Z^2-X^2-Y^2)+\frac{1}{r^5}\cdot(4Z)\\ &= \frac{Z}{r^5}\left(-\frac{5}{r^2}\cdot(2Z^2-X^2-Y^2-Z^2+Z^2)+4\right)\\ &= \frac{Z}{r^5}\left(-\frac{5}{r^2}\cdot(3Z^2-r^2)+4\right) = \frac{Z}{r^5}\left(-5\cdot\frac{3Z^2}{r^2}+5+4\right)\\ &= \frac{3Z}{r^5}\left(-5\cdot\frac{Z^2}{r^2}+3\right).\end{aligned}\tag{B.34}$$

Einsetzen von (B.32), (B.33) und (B.34) in Gleichung (B.31) liefert für den J_2-Störbeschleunigungsvektor schließlich die Beziehung

$$\nabla R_{2,0}(\boldsymbol{X}(t),\boldsymbol{p}) = \begin{pmatrix}-\frac{3}{2}\cdot GM\cdot\left(\frac{R_\mathrm{E}}{r}\right)^2\cdot C_{2,0}\cdot\frac{X}{r^3}\left(5\cdot\frac{Z^2}{r^2}-1\right)\\ -\frac{3}{2}\cdot GM\cdot\left(\frac{R_\mathrm{E}}{r}\right)^2\cdot C_{2,0}\cdot\frac{Y}{r^3}\left(5\cdot\frac{Z^2}{r^2}-1\right)\\ -\frac{3}{2}\cdot GM\cdot\left(\frac{R_\mathrm{E}}{r}\right)^2\cdot C_{2,0}\cdot\frac{Z}{r^3}\left(5\cdot\frac{Z^2}{r^2}-3\right)\end{pmatrix}.\tag{B.35}$$

Anhang C

C Zusätzliche Abbildungen

C.1 Prozessierung realer GRACE-Daten – Verarbeitung 4

Nachfolgend sind einige zusätzliche Grafiken gegeben zu der in Abschnitt 5.3.4, S. 113 beschriebenen vierten Prozessierung der realen GRACE-Daten.

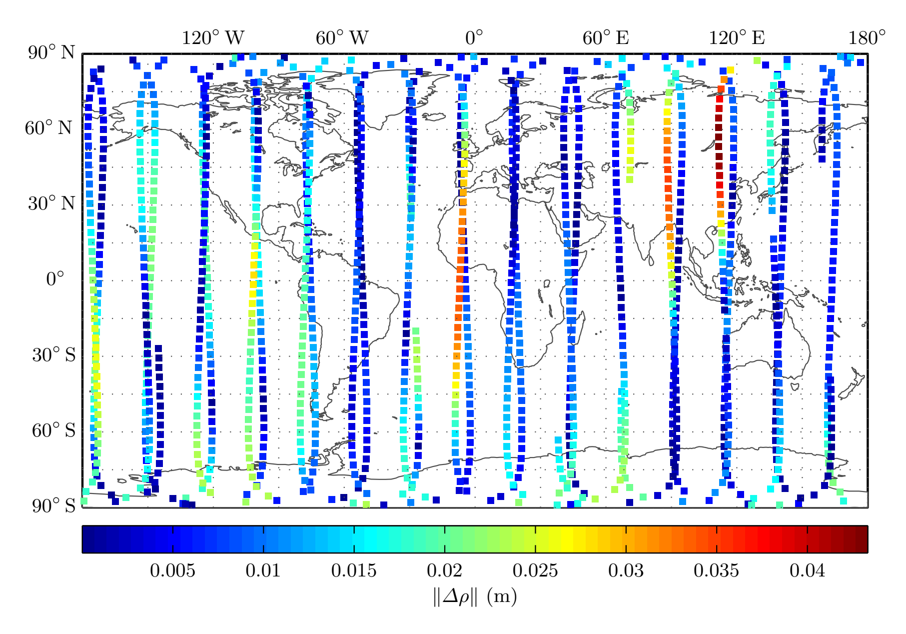


Abb. C.1: `fig17` – Navigationslösung vs. K-Band (Differenzen $\|\Delta\rho\|$)

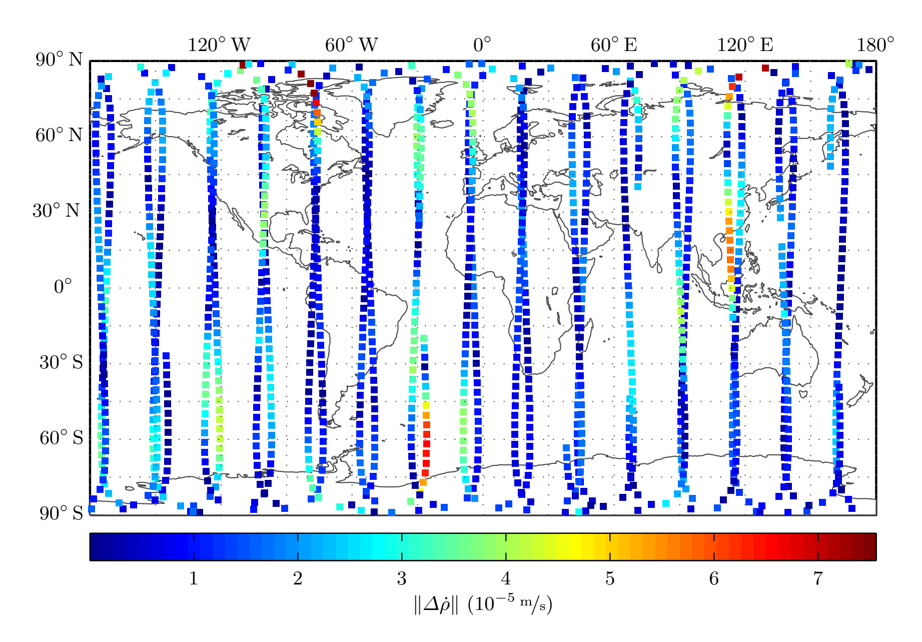


Abb. C.2: `fig18` – Navigationslösung vs. K-Band (Differenzen $\|\Delta\dot{\rho}\|$)

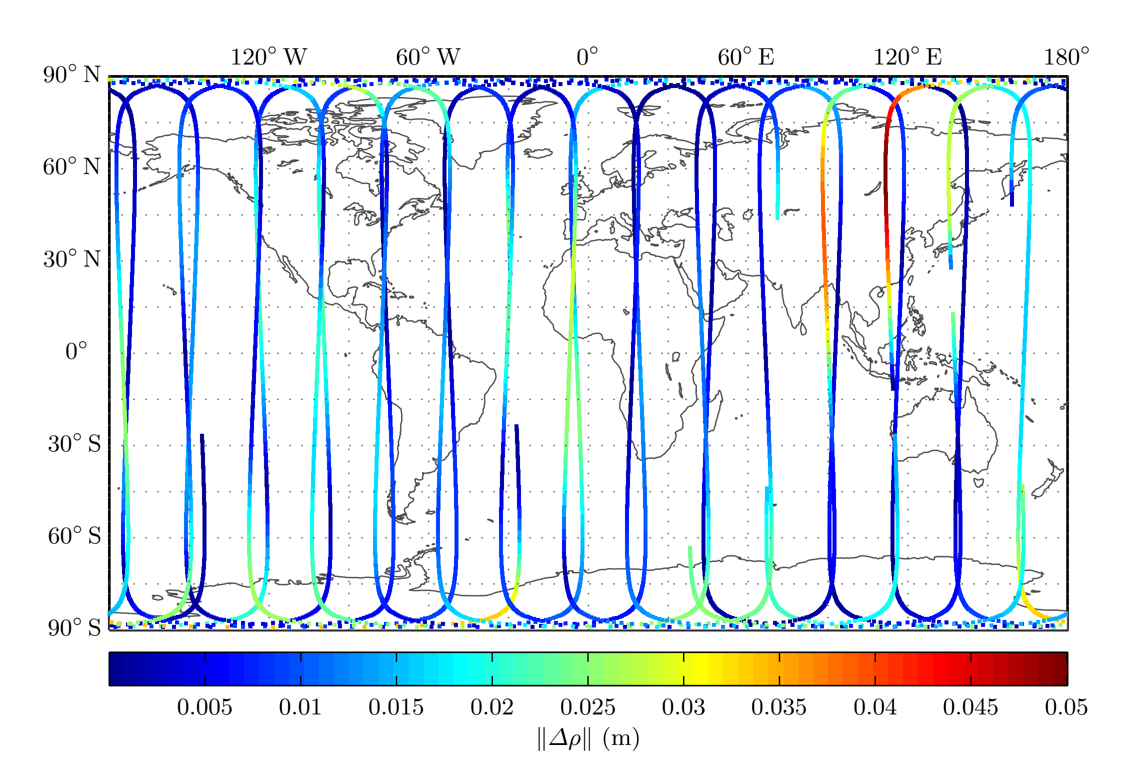


Abb. C.3: `fig19` – berechnete Orbits vs. K-Band (Differenzen $\|\varDelta\rho\|$)

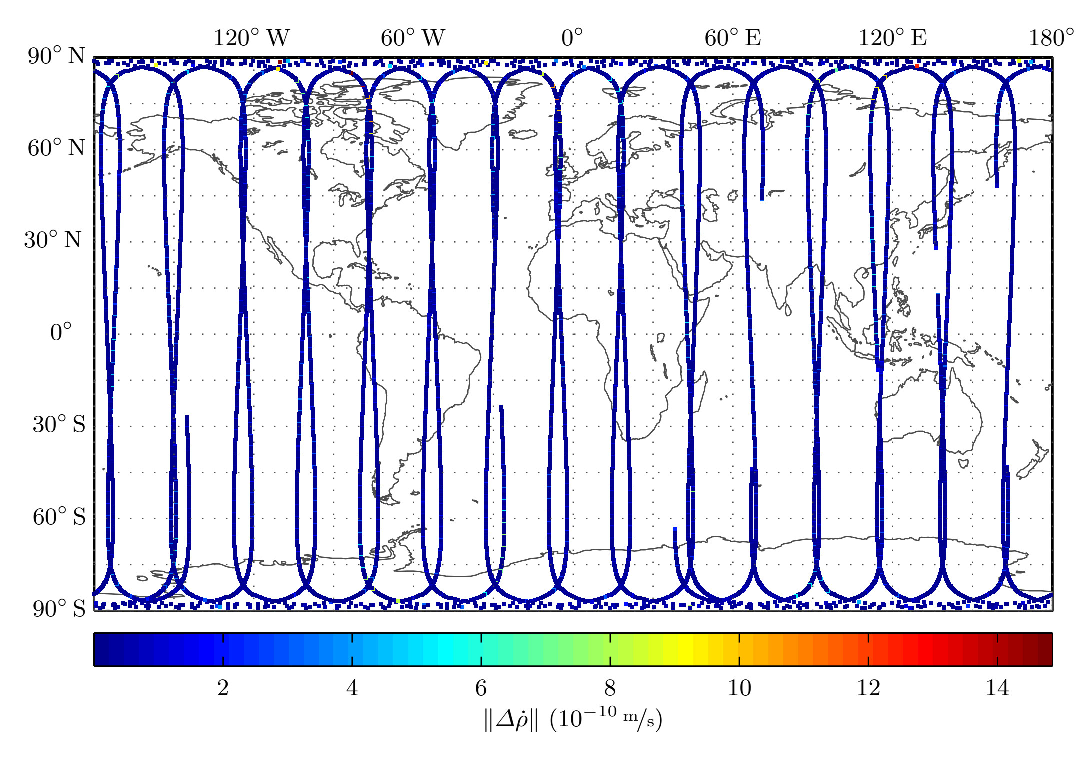


Abb. C.4: `fig20` – berechnete Orbits vs. K-Band (Differenzen $\|\varDelta\dot{\rho}\|$)

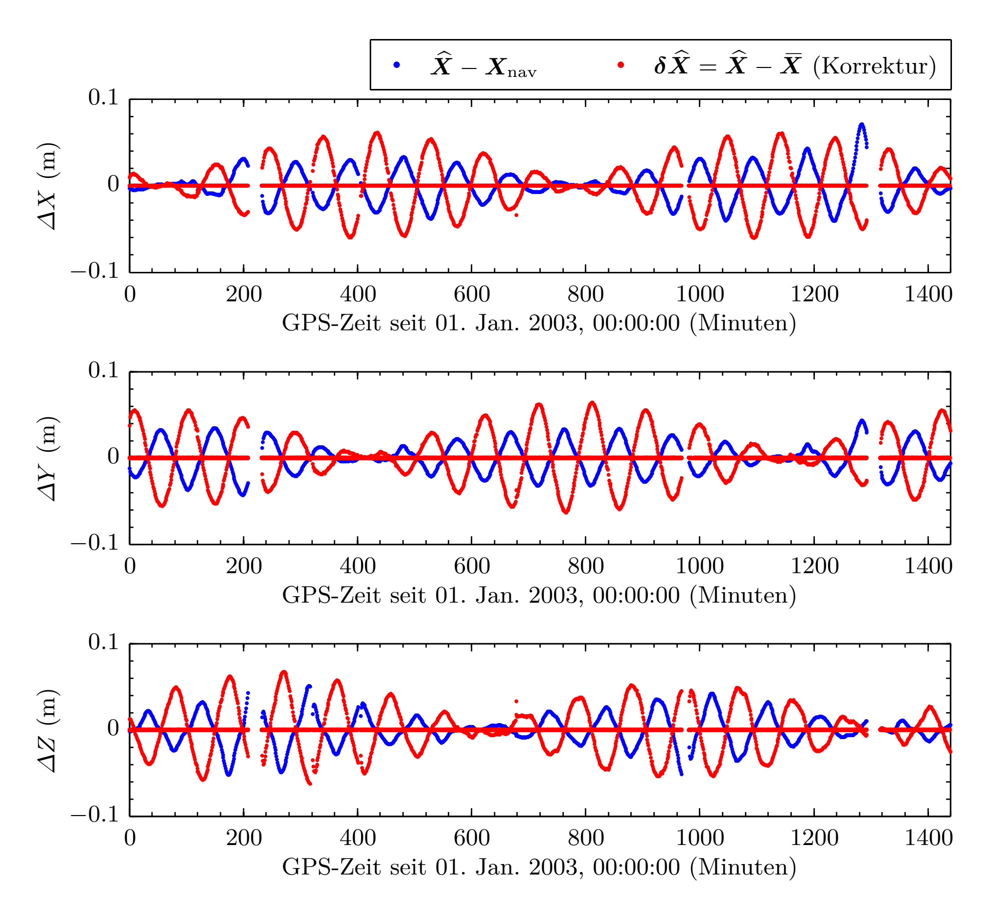


Abb. C.5: `fig03` – geschätzte Korrekturen und Vergleich zur Navigationslösung (Positionen, GRACE B)

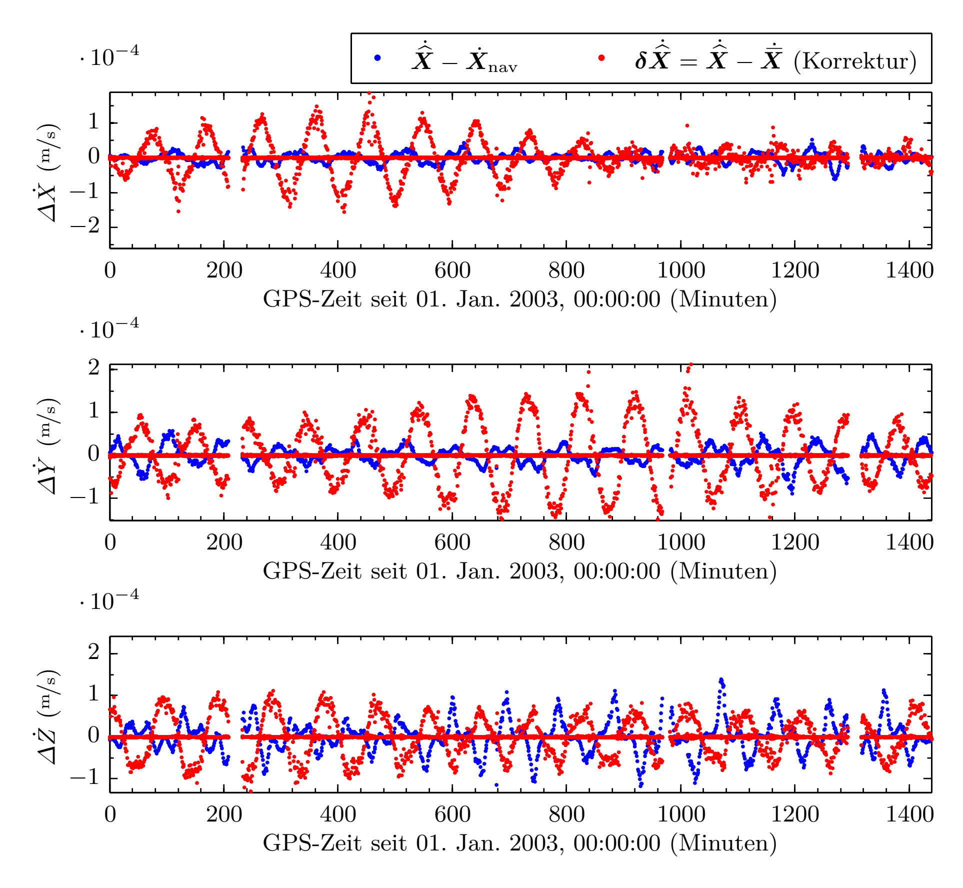


Abb. C.6: `fig03` – geschätzte Korrekturen u. Vergl. zur Navigationslösung (Geschwindigk., GRACE B)

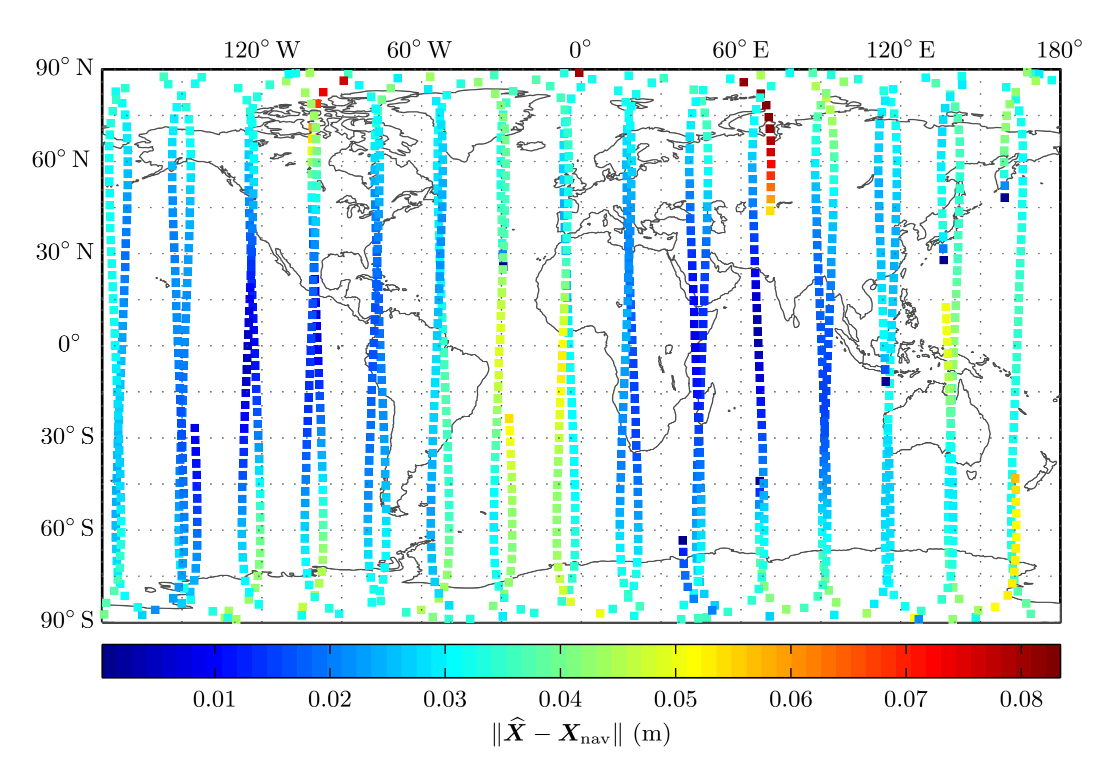


Abb. C.7: `fig13` – berechnete Positionen vs. Navigationslösung (GRACE B)

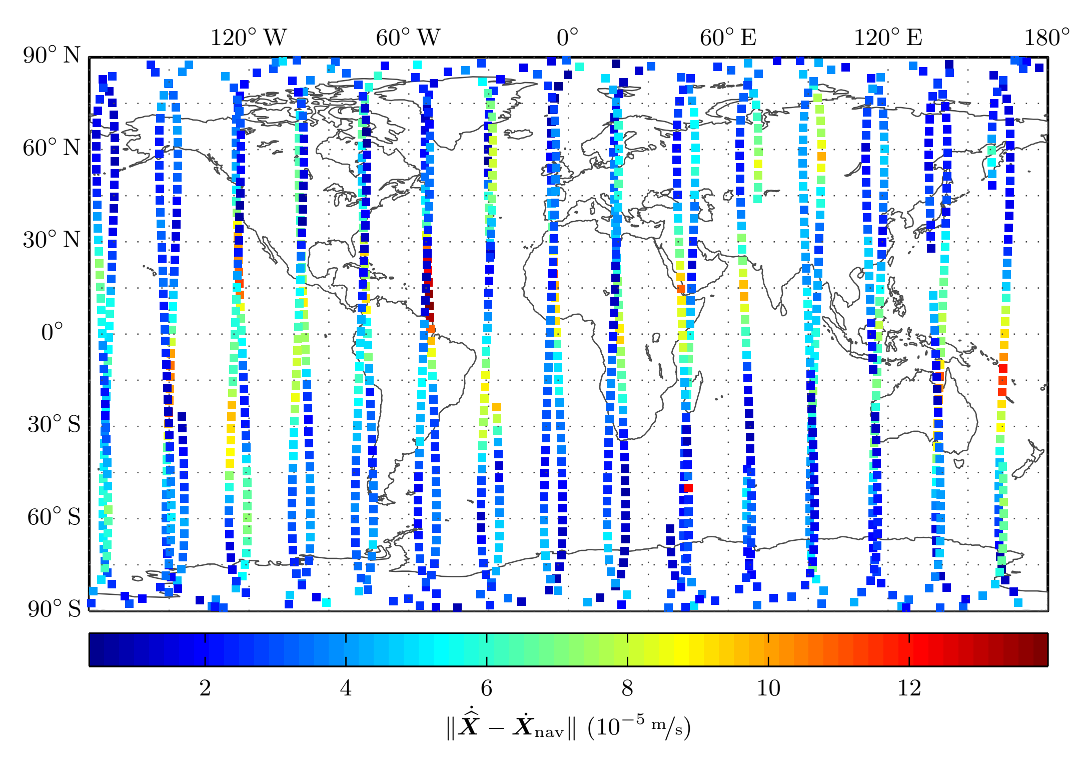


Abb. C.8: `fig14` – berechnete Geschwindigkeiten vs. Navigationslösung (GRACE B)

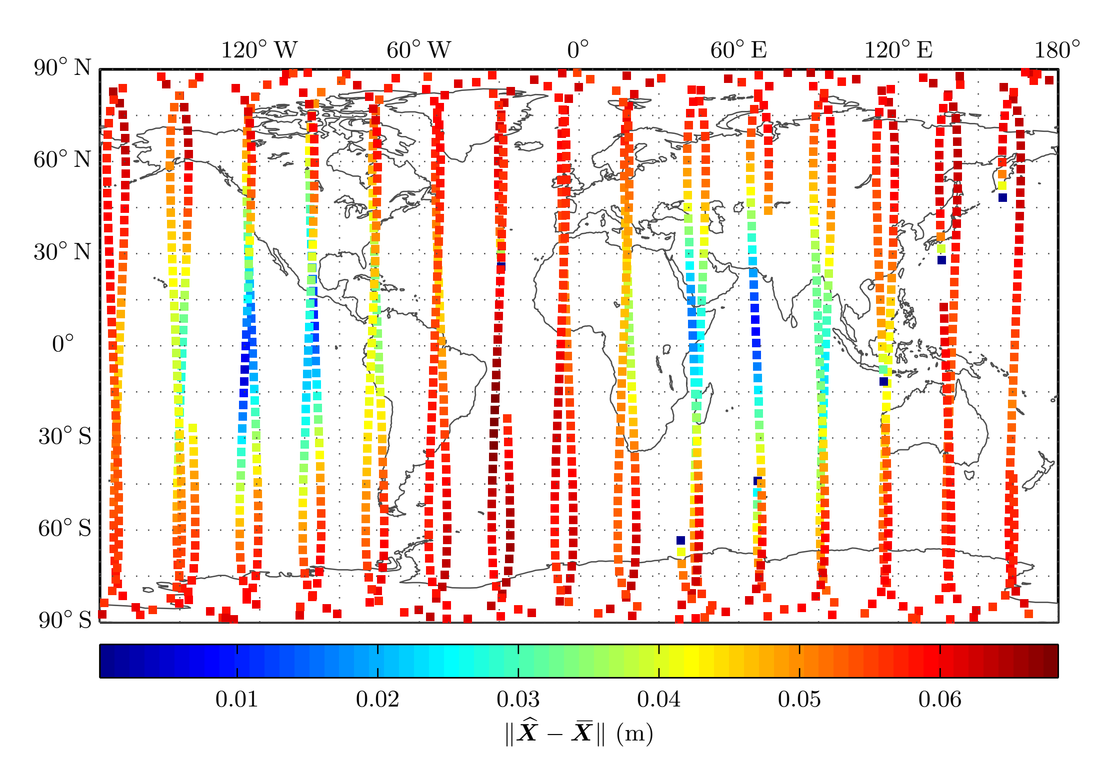


Abb. C.9: `fig15` – geschätzte Verbesserungen der prädizierten Positionen (GRACE B)

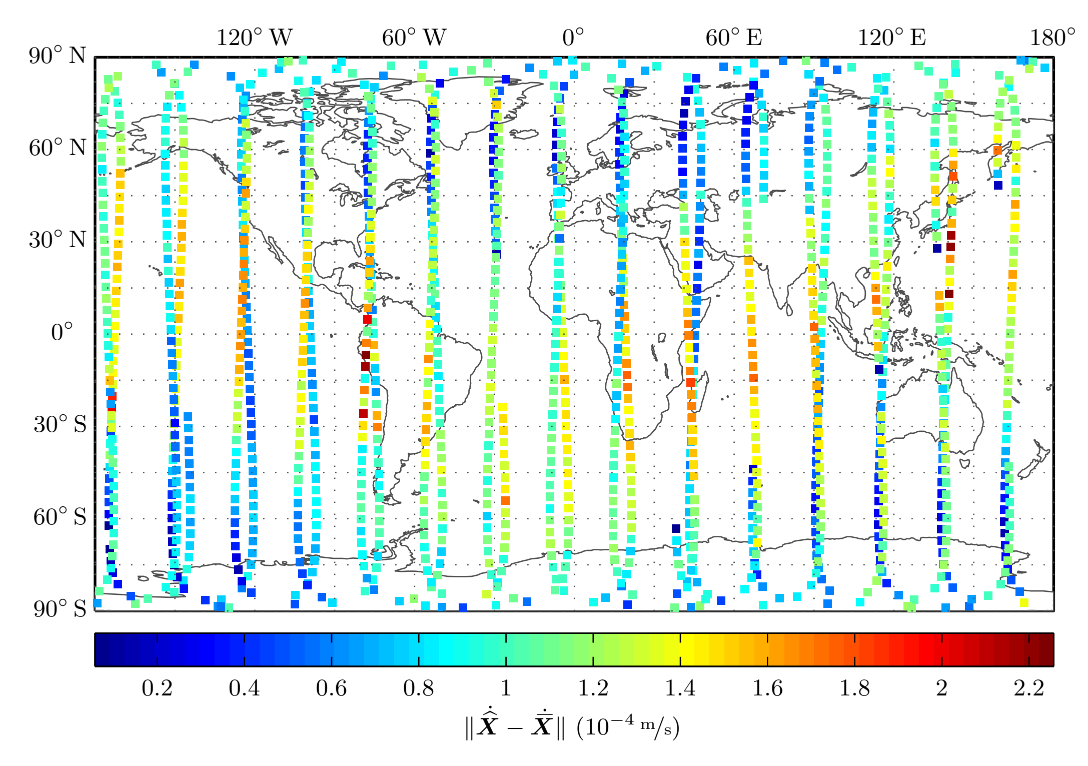


Abb. C.10: `fig16` – geschätzte Verbesserungen der prädizierten Geschwindigkeiten (GRACE B)

Literaturverzeichnis

AUSTEN, G.; GRAFAREND, E. W. (2004): *Gravitational field recovery from GRACE data of type high-low and low-low SST*. Proceedings of the Joint CHAMP/GRACE Science Meeting, 6-8 July 2004, GFZ Potsdam, http://www-app2.gfz-potsdam.de/pb1/JCG/Austen_and_Grafarend_jcg.pdf. [Zitiert auf Seiten 8 und 10]

AUSTEN, G.; REUBELT, T. (2000): *Räumliche Schwerefeldanalyse aus semi-kontinuierlichen Ephemeriden niedrigfliegender GPS-vermessener Satelliten vom Typ CHAMP, GRACE und GOCE*. Diplomarbeit, Geodätisches Institut der Universität Stuttgart. http://www.uni-stuttgart.de/gi/education/dipl/diplomarbeiten/austen+reubelt.pdf. [Zitiert auf Seiten 1 und 8]

BALLANI, L. (1988): *Partielle Ableitungen und Variationsgleichungen zur Modellierung von Satellitenbahnen und Parameterbestimmung*. In: Vermessungstechnik: geodätisch-kartographische Zeitschr. d. Deutschen Demokratischen Republik für Wissenschaft u. Praxis, Bd. 36(6), S. 192–194. [Zitiert auf Seite 38]

BALMINO, G.; RUMMEL, R.; VISSER, P.; WOODWORTH, P. L. (1999): *Gravity Field and Steady-State Ocean Circulation Mission: The Four Candidate Earth Explorer Core Missions*. ESA SP-1233(1), ESA Publications Division. http://esamultimedia.esa.int/docs/goce_sp1233_1.pdf. [Zitiert auf Seite 6]

BATTIN, R. H. (1999): *An introduction to the mathematics and methods of astrodynamics*. AIAA education series, Reston, Virginia, USA, überarb. Aufl. ISBN 1-56347-342-9. [Zitiert auf Seite 56]

BERGER, C.; BIANCALE, R.; ILL, M.; BARLIER, F. (1998): *Improvement of the empirical thermospheric model DTM: DTM94 – a comparative review of various temporal variations and prospects in space geodesy applications*. In: Journal of Geodesy, Bd. 72, S. 161–178. DOI: 10.1007/s001900050158. [Zitiert auf Seite 99]

BERTIGER, W.; DESAI, S. D.; HAINES, B.; HARVEY, N.; MOORE, A. W.; OWEN, S.; WEISS, J. P. (2010): *Single receiver phase ambiguity resolution with GPS data*. In: Journal of Geodesy, Bd. 84, S. 327–337. DOI: 10.1007/s00190-010-0371-9. [Zitiert auf Seiten VI und 100]

BERTIGER, W.; MEEHAN, T.; ROGSTAD, T.; NANDI, S.; BETTADPUR, S.; WU, S.-C.; DESAI, S.; WATKINS, M.; ROMANS, L.; KIM, J.; TIEN, J.; THOMAS, J. B.; HARRIS, I.; KRUIZINGA, G.; FRANKLIN, G.; NGUYEN, D.; KUANG, D.; DUNN, C.; HAINES, B. (2003): *Instrument of GRACE: GPS augments gravity measurements*. In: GPS World, Bd. 2(4), S. 16–28. [Zitiert auf Seite 6]

BETTADPUR, S. V. (2007): *GRACE Product Specification Document Rev. 4.5*. GRACE 327-720, Center for Space Research University of Texas at Austin (UTCSR), Jet Propulsion Laboratory (JPL). http://podaac.jpl.nasa.gov/grace/documentation.html. [Zitiert auf Seiten 7 und 99]

BODEN, D. G. (1986): *A comparison of nonlinear filters for orbit determination and estimation*. Doktorarbeit, University of Illinois, Urbana-Champaign. http://handle.dtic.mil/100.2/ADA170680. [Zitiert auf Seiten 39 und 133]

BROWN, R. G.; HWANG, P. Y. C. (1997): *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. Wiley, New York, 3. Aufl. ISBN 0-471-12839-2. [Zitiert auf Seiten 26 und 34]

BUCY, R. S.; JOSEPH, P. D. (1968): *Filtering for Stochastic Processes, with Applications to Guidance*. Wiley, New York. [Zitiert auf Seite 31]

CASE, K.; KRUIZINGA, G. L. H.; WU, S.-C. (2010): *GRACE Level 1B Data Product User Handbook v. 1.3*. GRACE 327-733, Jet Propulsion Laboratory (JPL). ftp://podaac.jpl.nasa.gov/allData/grace/docs/Handbook_1B_v1.3.pdf. [Zitiert auf Seiten 5, 99 und 132]

CHUI, C. K.; CHEN, G. (2009): *Kalman filtering with real time applications*. Springer series in information sciences. Springer, Berlin, 4. Aufl. ISBN 978-3-540-87849-0. DOI: 10.1007/978-3-540-87849-0. [Zitiert auf Seite 26]

Coleman, T. F.; Li, Y. (1992): *On the Convergence of Reflective Newton Methods for Large-scale Nonlinear Minimization Subject to Bounds*. Techn. Ber., Cornell University, Ithaca, NY, USA. http://hdl.handle.net/1813/6164. [Zitiert auf Seite 23]

Coleman, T. F.; Li, Y. (1996): *An interior trust region approach for nonlinear minimization subject to bounds*. In: SIAM Journal on Optimization, Bd. 6, S. 418–445. DOI: 10.1137/0806023. [Zitiert auf Seite 23]

Colombo, O. L. (1984): *The global mapping of gravity with two satellites*, *Publications on Geodesy*, Bd. 7(3). Netherlands Geodetic Commission, Delft. http://www.ncg.knaw.nl/Publicaties/Geodesy/28Colombo.html. [Zitiert auf Seiten 1, 8 und 10]

Curtis, H. D. (2010): *Orbital mechanics for engineering students*. Elsevier aerospace engineering series. Butterworth-Heinemann, 2. Aufl. ISBN 978-0-12-374778-5. [Zitiert auf Seiten 19 und 56]

Dever, C. W. (1998): *Vehicle model-based filtering for spacecraft attitude determination*. Thesis (S.M.), Massachusetts Institute of Technology, Dept. of Mechanical Engineering. http://hdl.handle.net/1721.1/47801. [Zitiert auf Seite 145]

Dunn, C.; Bertiger, W.; Franklin, G.; Harris, I.; Kruizinga, G.; Meehan, T.; Nandi, S.; Nguyen, D.; Rogstad, T.; Thomas, J. B.; Tien, J. (2002): *The instrument on NASA's GRACE Mission: Augmentation of GPS to achieve unprecedented gravity field measurements*. In: *Proceedings of ION GPS 2002*. Portland, OR, USA. http://hdl.handle.net/2014/10486. [Zitiert auf Seiten 13, 14 und 60]

Eichhorn, A. (2005): *Ein Beitrag zur Identifikation von dynamischen Strukturmodellen mit Methoden der adaptiven KALMAN-Filterung*. Doktorarbeit, Institut für Anwendungen der Geodäsie im Bauwesen, Universität Stuttgart, München. URN: urn:nbn:de:bsz:93-opus-23171. [Zitiert auf Seite 30]

Escobal, P. R. (1976): *Methods of orbit determination*. Robert E. Krieger Publishing Company, Malabar, Florida, 2. Aufl. ISBN 0-88275-319-3. [Zitiert auf Seite 55]

Eshagh, M. (2010): *Alternative expressions for gravity gradients in local north-oriented frame and tensor spherical harmonics*. In: Acta Geophysica, Bd. 58, S. 215–243. DOI: 10.2478/s11600-009-0048-z. [Zitiert auf Seite 51]

Freund, R. W.; Hoppe, R. H. W. (2007): *Stoer/Bulirsch: Numerische Mathematik 1*. Springer-Lehrbuch. Springer Berlin Heidelberg, 10. Aufl. ISBN 978-3-540-45389-5. DOI: 10.1007/978-3-540-45390-1_1. [Zitiert auf Seite 17]

Gelb, A. H. (Hg.) (1996): *Applied optimal estimation*. M.I.T. Press, Cambridge, Mass., 14. Aufl. ISBN 0-262-20027-9. [Zitiert auf Seiten 25, 26, 33, 37, 40, 41, 133 und 134]

Grafarend, E. W. (1994): *What is a Geoid?*. In: Vaníček, P.; Christou, N. T. (Hg.), *Geoid and its geophysical interpretations*, S. 3–32. CRC Press, Boca Raton, Florida. ISBN 0-8493-4227-9. [Zitiert auf Seite 1]

Grewal, M. S.; Andrews, A. P. (2008): *Kalman filtering: theory and practice using MATLAB*. Wiley, Hoboken, N.J, 3. Aufl. [Zitiert auf Seiten 25, 26, 35 und 133]

Götzelmann, M. (2003): *Short-Arc Bahnanpassung bei niedrig fliegenden Satelliten*. Diplomarbeit, Geodätisches Institut der Universität Stuttgart. http://www.uni-stuttgart.de/gi/education/dipl/diplomarbeiten/goetzelmann.pdf. [Zitiert auf Seiten 49, 50 und 72]

Gulick, L. J. (1970): *A comparison of methods for computing gravitational potential derivatives*. ESSA technical report C & GS 40, National Ocean Survey, National Oceanic and Atmospheric Administration, US. Department of Commerce, Rockville, Maryland. [Zitiert auf Seite 48]

Hanson, A. J. (2006): *Visualizing Quaternions (The Morgan Kaufmann Series in Interactive 3D Technology)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 0120884003. [Zitiert auf Seite 145]

Hartikainen, J.; Särkkä, S. (2008): *Optimal filtering with Kalman filters and smoothers – a Manual for Matlab toolbox EKF/UKF (v1.2)*. Department of Biomedical Engineering and Computational Science, Helsinki University of Technology (Espoo, Finland), http://www.lce.hut.fi/research/mm/ekfukf/documentation.pdf. [Zitiert auf Seite 134]

Henrici, P. (1962): *Discrete variable methods in ordinary differential equations*. Wiley, New York. ISBN 0-471-37224-2. [Zitiert auf Seite 72]

Hermann, M. (2004): *Numerik gewöhnlicher Differentialgleichungen: Anfangs- und Randwertprobleme*. Oldenbourg, München. ISBN 3-486-27606-9. [Zitiert auf Seiten 22 und 72]

Hess, D.; Keller, W. (1999): *Gradiometrie mit GRACE*. In: Zeitschrift für Vermessungswesen, Bd. 5, S. 137–144. [Zitiert auf Seiten 6, 10 und 11]

Hoffmann, A.; Marx, B.; Vogt, W. (2005): *Mathematik für Ingenieure*, Bd. 1: Lineare Algebra, Analysis, Theorie und Numerik. Pearson Studium, München. ISBN 3-8273-7113-9. [Zitiert auf Seiten 15 und 16]

Hofmann-Wellenhof, B.; Moritz, H. (2006): *Physical geodesy*. Springer, Wien, New York, 2., korr. Aufl. ISBN 3-211-33544-7. DOI: 10.1007/978-3-211-33545-1. [Zitiert auf Seite 46]

Ilk, K. (1983): *Formulierung von Energieaustauschbeziehungen zur Ausmessung des Gravitationsfeldes*. In: Schneider, M. (Hg.), *Die Arbeiten des Sonderforschungsbereiches 78 Satellitengeodäsie der Technischen Universität München im Jahre 1982, Veröffentlichungen der Bayerischen Kommission für die internationale Erdmessung, Astronomisch-geodätische Arbeiten*, 43, S. 128–166. Verlag der Bayerischen Akademie der Wissenschaften, München. ISBN 3-7696-9786-3. [Zitiert auf Seite 1]

Iske, A. (2007): *Skript zur Vorlesung „Analysis II“*. Im Sommersemester 2007 am Department Mathematik der Universität Hamburg, http://www.math.uni-hamburg.de/teaching/export/tuhh/cm/a2/index.html. [Zitiert auf Seite 17]

Jekeli, C. (1999): *The determination of gravitational potential differences from satellite-to-satellite tracking*. In: Celestial Mechanics and Dynamical Astronomy, Bd. 75, S. 85–101. DOI: 10.1023/A:1008313405488. [Zitiert auf Seite 8]

Jekeli, C.; Marson, I.; Sünkel, H. (1999): *An analysis of geopotential difference determination from satellite-to-satellite tracking*. In: *Proceedings of the 2nd Joint Meeting of the International Gravity Commission and the International Geoid Commission, 7-12 Sept. 1998, Trieste, Italy*, *Bollettino di Geofisica Teorica e Applicata*, Bd. 40(3-4), S. 267–272. Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS). [Zitiert auf Seiten 1 und 14]

Kalman, R. E. (1960): *A new approach to linear filtering and prediction problems*. In: Transactions of the ASME – Journal of basic engineering, Bd. 82 (Series D), S. 35–45. [Zitiert auf Seite 25]

Kalman, R. E.; Bucy, R. S. (1961): *New results in linear filtering and prediction theory*. In: Transactions of the ASME – Journal of basic engineering, Bd. 83 (Series D), S. 95–108. [Zitiert auf Seite 25]

Kaula, W. M. (1966): *Theory of satellite geodesy: applications of satellites to geodesy*. Blaisdell Publishing Company, Waltham, Massachusetts. [Zitiert auf Seite 19]

Keller, W.; Sharifi, M. A. (2005a): *GRACE Gradiometer*. In: *Gravity, Geoid and Space Missions*, *International Association of Geodesy Symposia*, Bd. 129, S. 42–47. Springer Berlin/Heidelberg. DOI: 10.1007/3-540-26932-0_8. [Zitiert auf Seite 10]

Keller, W.; Sharifi, M. A. (2005b): *Satellite gradiometry using a satellite pair*. In: Journal of Geodesy, Bd. 78(9), S. 544–557. DOI: 10.1007/s00190-004-0426-x. [Zitiert auf Seiten 9, 10, 11 und 14]

König, D.; Seitz, K. (2003): *Numerische Integration von Satellitenbahnen unter Berücksichtigung der Anisotropie des Gravitationsfeldes der Erde*. Interner Bericht, Schriftenreihe des Studiengangs Geodäsie und Geoinformatik, Universität Karlsruhe. Ersch. im Universitätsverlag Karlsruhe, URN: urn:nbn:de:0072-AAA53020037. [Zitiert auf Seiten 72 und 73]

Lemoine, F.; Kenyon, S.; Factor, J.; Trimmer, R.; Pavlis, N.; Chinn, D.; Cox, C.; Klosko, S.; Luthcke, S.; Torrence, M.; Wang, Y.; Williamson, R.; Pavlis, E.; Rapp, R.; Olson, L. J., T.R. Gulick (1998): *The development of the joint NASA GSFC and the National Imagery and Mapping Agency (NIMA) geopotential model EGM96*. NASA technical paper nasa/tp-1998-206861, Goddard Space Flight Center, Greenbelt, USA. [Zitiert auf Seite 47]

Loffeld, O. (1990): *Estimationstheorie I & II*. Einführung in die Nachrichtentechnik (Hrsg. Gottwald, A.). Oldenbourg, München. ISBN 3-486-21616-3; 3-486-21627-9. 2 Bde. in 662 S., Bd. 1: Grundlagen und stochastische Konzepte, Bd. 2: Anwendungen – Kalman-Filter. [Zitiert auf Seiten 25 und 26]

LYARD, F.; LEFEVRE, F.; LETELLIER, T.; FRANCIS, O. (2006): *Modelling the global ocean tides: modern insights from FES2004*. In: Ocean Dynamics, Bd. 56, S. 394–415. DOI: 10.1007/s10236-006-0086-x. [Zitiert auf Seite 99]

MAYBECK, P. S. (1979): *Stochastic models, estimation, and control*, *Mathematics in science and engineering*, Bd. 141-1. Academic Press, New York, London [u.a.]. ISBN 0-12-480701-1. [Zitiert auf Seiten 25, 26 und 37]

MAYBECK, P. S. (1982a): *Stochastic models, estimation, and control*, *Mathematics in science and engineering*, Bd. 141-2. Academic Press, New York, London [u.a.]. ISBN 0-12-480702-X. [Zitiert auf Seiten 26 und 133]

MAYBECK, P. S. (1982b): *Stochastic models, estimation, and control*, *Mathematics in science and engineering*, Bd. 141-3. Academic Press, New York, London [u.a.]. ISBN 0-12-480703-8. [Zitiert auf Seite 26]

MAYER-GÜRR, T. (2006): *Gravitationsfeldbestimmung aus der Analyse kurzer Bahnbögen am Beispiel der Satellitenmissionen CHAMP und GRACE*. Doktorarbeit, Inst. für Geodäsie und Geoinformation, Univ. Bonn. URN: urn:nbn:de:hbz:5N-09047. [Zitiert auf Seiten 1 und 8]

MAYER-GÜRR, T. (2007): *ITG-Grace03s: The latest GRACE gravity field solution computed in Bonn*. Presentation at GSTM+SPP, 15-17 Okt. 2007, Potsdam. http://www.igg.uni-bonn.de/apmg/fileadmin/DatenModelle/media/mayer-guerr_gstm_potsdam_2007.pdf. [Zitiert auf Seite 8]

MCGEE, L. A.; SCHMIDT, S. F. (1985): *Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry*. Technical memorandum 86847, National Aeronautics and Space Administration, Ames Research Center. http://hdl.handle.net/2060/19860003843 oder http://purl.access.gpo.gov/GPO/LPS104644. [Zitiert auf Seite 26]

MEYER, U. (2006): *Möglichkeiten und Grenzen der Hill-Gleichungen für die Schwerefeldbestimmung*. Doktorarbeit, Institut für Astronomische und Physikalische Geodäsie, Technische Universität München, Potsdam. DOI: 10.2312/GFZ.b103-06082. [Zitiert auf Seiten 49 und 50]

MONTENBRUCK, O.; GILL, E. (2005): *Satellite Orbits: Models, Methods and Applications; with 47 tables and a CD-ROM*. Springer, Berlin, korr. 3. Aufl. [Zitiert auf Seiten 21, 89 und 92]

MORÉ, J. J. (1978): *The Levenberg-Marquardt algorithm: Implementation and theory*. In: WATSON, G. A. (Hg.), *Numerical Analysis*, *Lecture Notes in Mathematics*, Bd. 630, S. 105–116. Springer Berlin/Heidelberg. ISBN 978-3-540-08538-6. DOI: 10.1007/BFb0067700. [Zitiert auf Seite 23]

NOVÁK, P.; AUSTEN, G.; SHARIFI, M. A.; GRAFAREND, E. W. (2006): *Mapping earth's gravitation using GRACE data*. In: *Observation of the Earth System from Space*, S. 149–164. Springer Berlin/Heidelberg. [Zitiert auf Seite 12]

O'KEEFE, J. A. (1957): *An application of Jacobi's integral to the motion of an Earth satellite*. In: Astronomical Journal, Bd. 62(1252), S. 265–266. DOI: 10.1086/107530. [Zitiert auf Seite 1]

O.T.-HANDBUCH (2010): *Optimization Toolbox™ User's Guide*. Online-Ressource. The MathWorks, Inc., http://www.mathworks.com/help/pdf_doc/optim/optim_tb.pdf. [Zitiert auf Seiten 23 und 128]

PARKINSON, C. L.; WARD, A.; KING, M. D. (Hg.) (2006): *Earth Science Reference Handbook – A Guide to NASA's Earth Science Program and Earth Observing Satellite Missions*. National Aeronautics and Space Administration, Washington, D.C., USA. http://eospso.gsfc.nasa.gov/ftp_docs/2006ReferenceHandbook.pdf. [Zitiert auf Seite 5]

PAVLIS, N.; HOLMES, S.; KENYON, S.; FACTOR, J. (2008): *An Earth Gravitational Model to Degree 2160: EGM2008*. Presentation at the 2008 General Assembly of the European Geosciences Union, Vienna, Austria, April 13-18, 2008. http://earth-info.nima.mil/GandG/wgs84/gravitymod/egm2008/NPavlis&al_EGU2008.ppt. [Zitiert auf Seite 47]

PETIT, G.; LUZUM, B. (2010): *IERS conventions (2010)*. IERS technical note no. 36, Verlag des Bundesamtes für Kartographie und Geodäsie, Frankfurt am Main. http://www.iers.org/IERS/EN/Publications/TechnicalNotes/tn36.html. [Zitiert auf Seiten 32, 47, 48 und 50]

PETROVSKAYA, M.; VERSHKOV, A. (2006): *Non-Singular Expressions for the Gravity Gradients in the Local North-Oriented and Orbital Reference Frames*. In: Journal of Geodesy, Bd. 80, S. 117–127. DOI: 10.1007/s00190-006-0031-2. [Zitiert auf Seiten 51, 65, 75, 87 und 143]

Rauch, H. E.; Tung, F.; Striebel, C. T. (1965): *Maximum likelihood estimates of linear dynamic systems*. In: AIAA Journal, Bd. 3(8), S. 1445–1450. DOI: 10.2514/3.3166. [Zitiert auf Seiten 62 und 134]

Reed, G. B. (1973): *Application of kinematical geodesy for determining the short wave length components of the gravity field by satellite gradiometry*. Reports of the department of geodetic science, report no. 201, Ohio State University, Research Foundation, Columbus, Ohio. [Zitiert auf Seiten 48, 49 und 50]

Reigber, C.; Balmino, G.; Schwintzer, P.; Biancale, R.; Bode, A.; Lemoine, J.-M.; König, R.; Loyer, S.; Neumayer, H.; Marty, J.-C.; Barthelmes, F.; Perosanz, F.; Zhu, S. Y. (2003): *Global gravity field recovery using solely GPS tracking and accelerometer data from CHAMP*. In: Space Science Reviews, Bd. 108(1-2), S. 55–66. [Zitiert auf Seite 6]

Reigber, C.; Jochmann, H.; Wünsch, J.; Petrovic, S.; Schwintzer, P.; Barthelmes, F.; Neumayer, K.-H.; König, R.; Förste, C.; Balmino, G.; Biancale, R.; Lemoine, J.-M.; Loyer, S.; Perosanz, F. (2005a): *Earth gravity field and seasonal variability from CHAMP*. In: *Earth Observation with CHAMP – results from three years in orbit*, S. 25–30. Springer Berlin/Heidelberg. [Zitiert auf Seite 6]

Reigber, C.; Schmidt, R.; Flechtner, F.; König, R.; Meyer, U.; Neumayer, K. H.; Schwintzer, P.; Zhu, S. Y. (2005b): *An Earth gravity field model complete to degree and order 150 from GRACE: EIGEN-GRACE02S*. In: Journal of Geodynamics, Bd. 39(1), S. 1–10. [Zitiert auf Seite 7]

Reinhardt, H.-J. (2008): *Numerik gewöhnlicher Differentialgleichungen: Anfangs- und Randwertprobleme*. de Gruyter, Berlin. ISBN 978-3-11-020033-1. [Zitiert auf Seiten 22, 23 und 72]

Reubelt, T. (2009): *Harmonische Gravitationsfeldanalyse aus GPS-vermessenen kinematischen Bahnen niedrig fliegender Satelliten vom Typ CHAMP, GRACE und GOCE mit einem hoch auflösenden Beschleunigungsansatz*. Doktorarbeit, Geodätisches Institut der Universität Stuttgart. URN: urn:nbn:de:bsz:93-opus-39258. [Zitiert auf Seiten 8, 21 und 48]

Rummel, R. (2003): *How to climb the gravity wall*. In: Space Science Reviews, Bd. 108(1-2), S. 1–14. [Zitiert auf Seite 12]

Rummel, R.; Ilk, K. H.; Reigber, C. (1978): *The use of satellite-to-satellite tracking for gravity parameter recovery*. In: *Proceedings of the European Workshop on Space Oceanography, Navigation and Geodynamics (SONG) held at Schloss Elmau, 16-21 January 1978 (ESA SP-137)*, S. 153–161. [Zitiert auf Seiten 1, 8 und 9]

Schaub, H.; Junkins, J. L. (2003): *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston, Virginia, USA, 1. Aufl. ISBN 1-56347-563-4. [Zitiert auf Seiten 55 und 56]

Scheinert, M. (1996): *Zur Bahndynamik niedrigfliegender Satelliten*. Doktorarbeit, Fakultät für Bauingenieur- und Vermessungswesen, Universität Stuttgart, München. [Zitiert auf Seiten 48, 49, 50, 51 und 72]

Schäfer, C. (2000): *Space gravity spectroscopy: the sensitivity analysis of GPS-tracked satellite missions (case study CHAMP)*. Doktorarbeit, Geodätisches Institut der Universität Stuttgart. URN: urn:nbn:de:bsz:93-opus-7050. [Zitiert auf Seiten 26 und 49]

Schlotterer, M. (2008): *Robuste Schätzung und Sensorfusion zur Navigation von wiederverwendbaren Raumtransportern*. Doktorarbeit, Universität Bremen, Fachbereich Produktionstechnik, Göttingen. [Zitiert auf Seite 40]

Schmidt, R. (1999): *Numerische Integration gestörter Satellitenbahnen mit MATLAB*. IAPG/FESG-Schriftenreihe Nr. 5, Diplomarbeit am Institut für astronomische und physikalische Geodäsie, Forschungseinrichtung Satellitengeodäsie, TU München. http://www.iapg.bv.tum.de/Schriftenreihe/. [Zitiert auf Seiten 48, 72, 73 und 86]

Schneeweiss, W. G. (1974): *Zufallsprozesse in dynamischen Systemen*. Ingenieurwissenschaftliche Bibliothek (Hrsg. Szabó, I.). Springer Berlin/Heidelberg/New York. ISBN 3-540-06192-4. [Zitiert auf Seiten 25, 26 und 28]

Schrick, K.-W. (Hg.) (1977): *Anwendungen der Kalman-Filter-Technik: Anleitung und Beispiele*. Methoden der Regelungstechnik. Oldenbourg, München [u.a.]. ISBN 3-486-20931-0. [Zitiert auf Seiten 25, 26, 30, 38 und 40]

Seeber, G. (1989): *Satellitengeodäsie: Grundlagen, Methoden u. Anwendungen*. de Gruyter, Berlin. ISBN 3-11-010082-7. [Zitiert auf Seiten 18, 19, 20, 43, 46, 85 und 148]

Shampine, L. F.; Gladwell, I.; Thompson, S. (2003): *Solving ODEs with MATLAB*. Cambridge Univ. Press, Cambridge, 1. Aufl. ISBN 9780521530941. DOI: 10.2277/0521530946. [Zitiert auf Seite 72]

Shampine, L. F.; Reichelt, M. W. (1997): *The MATLAB ODE suite*. In: SIAM Journal on Scientific Computing, Bd. 18(1), S. 1–22. DOI: 10.1137/S1064827594276424, auch verfügbar unter http://www.mathworks.com/help/pdf_doc/otherdocs/ode_suite.pdf. [Zitiert auf Seite 72]

Sharifi, M. A. (2004): *Satellite gradiometry using a satellite pair*. Diplomarbeit, Geodätisches Institut der Universität Stuttgart. URN: urn:nbn:de:bsz:93-opus-19980. [Zitiert auf Seite 10]

Shoemake, K. (1985): *Animating rotation with quaternion curves*. In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’85, S. 245–254. ACM, New York, USA. ISBN 0-89791-166-0. DOI: 10.1145/325334.325242. [Zitiert auf Seite 146]

Shuster, M. D. (1993): *A survey of attitude representations*. In: Journal of the Astronautical Sciences, Bd. 41, S. 439–517. http://home.comcast.net/~mdshuster/Pub_1993h_J_Repsurv_scan.pdf. [Zitiert auf Seiten 145 und 147]

Simon, D. (2006): *Optimal state estimation: Kalman, H. [infinity] and nonlinear approaches*. Wiley-Interscience, Hoboken, NJ. ISBN 0-471-70858-5. [Zitiert auf Seiten 26, 28, 31, 33, 37, 62 und 133]

Sneeuw, N. (2006): *Skript zur Vorlesung „Dynamic Satellite Geodesy“ (Analytische Bahnberechnung künstlicher Satelliten)*. Geodätisches Institut d. Univ. Stuttgart, http://www.uni-stuttgart.de/gi/education/dipl/lecturenotes.de.html. [Zitiert auf Seiten 52 und 148]

Sneeuw, N.; Schaub, H. (2005): *Satellite clusters for future gravity field missions*. In: *Gravity, Geoid and Space Missions*, International Association of Geodesy Symposia, Bd. 129, S. 12–17. Springer Berlin/Heidelberg. [Zitiert auf Seite 11]

Sneeuw, N.; Sharifi, M.; Keller, W. (2008): *Gravity Recovery from Formation Flight Missions*. In: *VI Hotine-Marussi Symposium on Theoretical and Computational Geodesy*, International Association of Geodesy Symposia, Bd. 132, S. 29–34. Springer Berlin/Heidelberg. [Zitiert auf Seite 11]

Särkkä, S. (2006): *Recursive Bayesian Inference on Stochastic Differential Equations*. Dissertation for the degree of Doctor of Science in Technology, Helsinki University of Technology (Espoo, Finland), Laboratory of Computational Engineering. http://lib.tkk.fi/Diss/2006/isbn9512281279/. [Zitiert auf Seiten 133 und 134]

Stoer, J.; Bulirsch, R. (Hg.) (2005): *Numerische Mathematik 2*. Springer-Lehrbuch. Springer, Berlin/Heidelberg, 5. Aufl. ISBN 978-3-540-26268-8. Online-Ressource: DOI: 10.1007/b137272. [Zitiert auf Seite 72]

Talman, R. (Hg.) (2007): *Geometric mechanics: toward a unification of classical physics*. WILEY-VCH, Weinheim, 2. Aufl. ISBN 978-3-527-40683-8. DOI: 10.1002/9783527611409. [Zitiert auf Seite 148]

Tapley, B.; Ries, J.; Bettadpur, S.; Chambers, D.; Cheng, M.; Condi, F.; Gunter, B.; Kang, Z.; Nagel, P.; Pastor, R.; Pekker, T.; Poole, S.; Wang, F. (2005): *GGM02 – An improved Earth gravity field model from GRACE*. In: Journal of Geodesy, Bd. 79(8), S. 467–478. http://www.csr.utexas.edu/grace/gravity/ggm02/GGM02_JGeod_190_0480.pdf. [Zitiert auf Seiten 7 und 47]

Tapley, B. D.; Watkins, M. M.; Ries, J. C.; Davis, G. W.; Eanes, R. J.; Poole, S. R.; Rim, H. J.; Schutz, B. E.; Shum, C. K.; Nerem, R. S.; Lerch, F. J.; Marshall, J. A.; Klosko, S. M.; Pavlis, N. K.; Williamson, R. G. (1996): *The Joint Gravity Model 3*. In: Journal of Geophysical Research, Bd. 101(B12), S. 28029–28049. DOI: 10.1029/96JB01645, auch verfügbar unter http://geodesy.eng.ohio-state.edu/course/refpapers/. [Zitiert auf Seite 47]

Torge, W. (2001): *Geodesy*. de Gruyter, Berlin, New York, 3. Aufl. ISBN 3-11-017072-8. [Zitiert auf Seite 47]

Vallado, D. A.; McClain, W. D. (2007): *Fundamentals of astrodynamics and applications*. Space technology library. Microcosm Press/Springer, Hawthorne, Calif./New York, 3. Aufl. ISBN 978-0-387-71831-6. [Zitiert auf Seiten 19, 35, 37 und 38]

VERMEER, M. (2010): *Methods of navigation*. Lecture notes, Department of Surveying, Aalto University School of Science and Technology, http://users.tkk.fi/~mvermeer/nav_en.pdf. [Zitiert auf Seite 31]

WEIGELT, M. L.; KELLER, W. (2008): *Line-of-sight Gradiometrie und ihre praktische Umsetzung im Falle von GRACE*. Geodätische Woche 30.9.–2.10.2008 in Bremen (Poster). http://www.uni-stuttgart.de/gi/research/paper/Weigelt_etal_GWBremen.pdf. [Zitiert auf Seite 14]

WENDEL, J. (2007): *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. Oldenbourg, München, 1. Aufl. ISBN 3-486-58160-0. [Zitiert auf Seite 30]

WIRSCHING, G. J. (Hg.) (2006): *Gewöhnliche Differentialgleichungen: Eine Einführung mit Beispielen, Aufgaben und Musterlösungen*. B. G. Teubner Verlag, GWV Fachverlage GmbH, Wiesbaden. ISBN 978-3-8351-9044-3. DOI: 10.1007/978-3-8351-9044-3. [Zitiert auf Seite 37]

WOLFF, M. (1969): *Direct measurements of the earth's gravitational potential using a satellite pair*. In: Journal of Geophysical Research, Bd. 74(22), S. 5295–5300. [Zitiert auf Seite 7]

WRIGHT, J. R. (1981): *Sequential orbit determination with auto-correlated gravity modeling errors*. In: Journal of Guidance and Control, Bd. 4(3), S. 304–309. AIAA paper 80-0239R, DOI: 10.2514/3.56083. [Zitiert auf Seite 39]

WRIGHT, J. R.; WOODBURN, J.; TRUONG, S.; CHUBA, W. (2008): *Orbit gravity error covariance*. 18th AAS/AIAA Space Flight Mechanics Meeting, Paper AAS 08-157, Galveston, Texas. http://www.stk.com/resources/user-resources/downloads/white-papers.aspx. [Zitiert auf Seiten 39, 42 und 133]

WU, S.-C.; KRUIZINGA, G.; BERTIGER, W. I. (2006): *Algorithm Theoretical Basis Document for GRACE Level 1B Data Processing v. 1.2*. GRACE 327-741, Jet Propulsion Laboratory (JPL). http://podaac.jpl.nasa.gov/grace/documentation.html. [Zitiert auf Seiten 99 und 132]